Cartographie RGB-D dense pour la localisation visuelle temps-réel et la navigation autonome

Maxime Meilland

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

MINES ParisTech

École doctorale n°84 :
Sciences et technologies de l'information et de la communication

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l'École nationale supérieure des mines de Paris

Spécialité « Informatique temps-réel, robotique et automatique »

présentée et soutenue publiquement par

Maxime MEILLAND

le 28 mars 2012

Cartographie RGB-D dense pour la localisation visuelle temps-réel et la navigation autonome

Directeur de thèse : **Patrick RIVES**
Co-encadrement de la thèse : **Andrew I. COMPORT**

Jury

Eric MARCHAND, Professeur, INRIA, Université de Rennes 1 (Hdr)	Rapporteur
Youcef MEZOUAR, Maitre de conférence, LASMEA, Université Blaise Pascal (Hdr)	Rapporteur
Vincent LEPETIT, Senior Researcher, CVLab, EPFL	Examinateur
Jean-Paul MARMORAT, Directeur de recherche, CMA, Ecole des Mines de Paris	Examinateur
Patrick RIVES, Directeur de recherche, AROLAG, INRIA Sophia Antipolis Méditerranée	Examinateur
Andrew COMPORT, Chargé de recherche CNRS, I3S, Université de Nice	Examinateur

T H È S E

Inria Sophia Antipolis - Méditerranée
2004, route des Lucioles - BP 93
06902 Sophia Antipolis Cedex, France

À mes parents.

Remerciements

Ces trois années de doctorat ont été pour ma part extrêmement enrichissantes. D'abord scientifiquement, car elles m'ont permis d'aborder une thématique variée, mais aussi humainement, car j'ai pu échanger avec des personnes venant d'horizons et de cultures différentes, toutes passionnées par leur métier, par l'envie d'apprendre et le désir d'innover.

Je tiens tout d'abord à remercier monsieur Jean-Paul Marmorat pour avoir accepté de présider mon jury de thèse, messieurs Eric Marchand et Youcef Mezouar pour leur travail de rapporteurs et monsieur Vincent Lepetit pour avoir examiné mon manuscrit.

Je tiens également à remercier chaleureusement mes encadrants : Patrick Rives pour m'avoir permis d'effectuer cette thèse à l'INRIA Sophia Antipolis, dans un cadre de travail excellent, et aussi pour sa grande expérience et disponibilité. Andrew Comport, pour m'avoir guidé tout au long de ce travail, pour son enthousiasme et ses idées toujours plus innovantes, ainsi que pour ses relectures d'articles qui m'ont permis de progresser dans mes travaux.

J'aimerai aussi remercier tous mes collègues et anciens collègues côtoyés au sein de l'équipe Arobas : Claude, Pascal, Ezio, Adrien, Nathalie, Cyril, Gabriella, Thomas, Patrick, Tiago, Minh Duc, Wladyslaw, Alexandre, Tawsif, Stefan, Mélaine, Mathieu, Claire et plus spécialement Glauco, Daniele, Luca et Adan pour leur amitié et tous les bons moments passés ensemble que je ne suis pas près d'oublier.

Enfin je remercie mes parents et ma famille en général, pour m'avoir toujours encouragé à poursuivre le cursus que j'ai choisi. Un merci spécial à Silvia et Gatto, pour leur soutien affectif de tous les instants.

Table des matières

Table des matières 3

Introduction et notations 3

I Localisation 11

1 État de l'art 13
1.1 Introduction . . . 13
1.2 Estimation du mouvement d'une caméra . . . 14
1.2.1 Méthodes basées points d'intérêt . . . 14
1.2.2 Méthodes directes . . . 15
1.3 *SLAM* . . . 16
1.4 Méthodes avec apprentissage . . . 17
1.4.1 Modèles 3D . . . 17
1.4.2 Mémoires image . . . 19
1.5 Conclusion . . . 19

2 Suivi visuel 3D direct 21
2.1 Introduction . . . 21
2.2 Notions de géométrie . . . 21
2.2.1 Transformation rigide . . . 21
2.2.2 Transformation de vitesse . . . 22
2.3 Formation de l'image . . . 23
2.3.1 Projection perspective . . . 23
2.3.2 Distorsions radiales . . . 23
2.4 Transformation d'une image . . . 24
2.4.1 Fonction de *warping* . . . 24
2.4.2 Interpolations . . . 25
2.4.3 Hypothèse Lambertienne . . . 27
2.5 Fonction d'erreur : SSD . . . 27
2.6 Minimisation efficace . . . 27
2.6.1 Approximation du système d'équations . . . 27
2.6.2 Minimisation . . . 28
2.6.3 Inverse compositionnelle (IC) . . . 29
2.6.4 Approximation du second ordre (ESM) . . . 30

2.6.5 M-estimateurs . . . 31
2.7 Pyramide multi-résolution . . . 32
2.8 Conclusion . . . 34

II Modélisation dense de l'environnement 37

3 Construction d'une sphère augmentée 43
3.1 Systèmes existants . . . 43
3.1.1 Sphère photométrique panoramique . . . 43
3.1.2 Sphère de profondeur . . . 46
3.2 Système d'acquisition de sphères augmentées . . . 48
3.2.1 Système de caméras à *multi-baselines* . . . 48
3.2.2 Étalonnage . . . 49
3.2.3 Extraction de la profondeur . . . 51
3.2.4 Fusion de l'information . . . 55
3.2.5 Échantillonnage d'une sphère . . . 57
3.3 Conclusion . . . 58

4 Cartographie 61
4.1 Introduction . . . 61
4.2 Odométrie visuelle sphérique 3D . . . 61
4.2.1 Modélisation du problème . . . 61
4.2.2 Minimisation globale . . . 62
4.3 Sélection automatique des sphères du graphe . . . 64
4.4 Résultats expérimentaux . . . 65
4.4.1 Positionnement des sphères . . . 65
4.4.2 Navigation virtuelle photo-réaliste . . . 66
4.5 Conclusion . . . 67

5 Sélection d'information 69
5.1 Introduction . . . 69
5.2 État de l'art . . . 69
5.3 Sélection de pixels saillants . . . 71
5.3.1 Gradients géométriques et photométriques . . . 71
5.3.2 Résultats de simulations . . . 76
5.4 Conclusion . . . 78

III Localisation en ligne et navigation autonome 81

6 Localisation temps réel 83
6.1 Introduction . . . 83
6.2 Localisation . . . 83
6.3 Sélection de l'image de référence . . . 84
6.3.1 Cas d'un graphe d'images sphériques . . . 84
6.3.2 Cas d'un graphe d'images perspectives . . . 84

6.4 Estimation efficace de la rotation 3D locale . . . 86
6.5 Utilisation de plusieurs nœuds du graphe . . . 87
6.5.1 Comparaison mono/multi modèle . . . 88
6.6 Résultats . . . 88
6.6.1 Implémentation . . . 88
6.6.2 Initialisation . . . 90
6.6.3 Environnements urbains . . . 91
6.6.4 Environnement intérieur . . . 92
6.7 Conclusion . . . 95

7 Robustesse aux changements d'illumination 99
7.1 Introduction . . . 99
7.2 État de l'art . . . 99
7.3 Suivi hybride . . . 101
7.3.1 Modèle d'illumination biais global, gain local . . . 101
7.3.2 Suivi basé modèle . . . 101
7.3.3 Suivi basé odométrie visuelle . . . 102
7.3.4 Optimisation globale . . . 103
7.3.5 Exemple de localisation hybride . . . 104
7.4 Résultats expérimentaux . . . 107
7.4.1 Comparaison des techniques . . . 107
7.4.2 Robustesse . . . 108
7.5 Conclusion . . . 109

8 Navigation autonome 113
8.1 Présentation du système . . . 113
8.2 Suivi de trajectoire . . . 113
8.2.1 Modèle cinématique du véhicule . . . 113
8.2.2 Trajectoire de consigne . . . 116
8.2.3 Loi de commande . . . 116
8.3 Détection d'obstacles . . . 118
8.4 Station déportée . . . 118
8.5 Résultats expérimentaux . . . 118
8.5.1 Sophia Antipolis . . . 118
8.5.2 Clermont Ferrand . . . 118
8.6 Conclusion . . . 123

IV Conclusion et perspectives 125

V Annexes 129

Calcul des matrices Jacobiennes 129

DECSlam 129

Références **149**

Table des figures

1.1 Comparaison des méthodes d'estimation itératives. 15
1.2 Modèle 3D texturé. 18

2.1 Transformation rigide 22
2.2 Position des caméras dans la scène. 25
2.3 Interpolation bilinéaire. 26
2.4 Processus de minimisation directe robuste. 33
2.5 Pyramide multi-résolution. 35
2.6 Représentation locale : sphère augmentée $\boldsymbol{\mathcal{S}}$. 39
2.7 Sphères multi-résolution. 40
2.8 Représentation égo-centrée : graphe de sphères augmentées $\mathcal{G}$. 41

3.1 Objectif grand angle et image fisheye. 44
3.2 Caméra omnidirectionnelle catadioptrique. 44
3.3 Modèle de projection unifié. 45
3.4 Systèmes multi-caméra. 47
3.5 Système d'acquisition de sphères augmentées. 49
3.6 Étalonnage du système d'acquisition sphérique. 50
3.7 Étalonnage du système d'acquisition : évolution de l'erreur. 51
3.8 Repères de rectification stéréo. 52
3.9 Rectification d'une paire d'images stéréo. 54
3.10 Projection sphérique. 56
3.11 Images sphériques reconstruites avec le système à *baseline*. 59
3.12 Échantillonnage d'une sphère unitaire. 60

4.1 Odométrie visuelle sphérique 3D. 63
4.2 Évolution de la MAD en fonction de la distance parcourue. 65
4.3 Graphe de sphères couvrant 1.5 kms. 66
4.4 Image de synthèse générée à partir d'une sphère augmentée. 67

5.1 Exemple de façades à "l'infini". 71
5.2 Exemple de façades à forts gradients. 72
5.3 Cartes de saillance associées aux 6 degrés de liberté. 73
5.4 Sélection des meilleurs pixels. 74
5.5 Saillance multi-résolution. 75
5.6 Simulation d'un plan à l'infini. 76
5.7 Comparaison des méthodes : dense, GP et GP+GG 77

5.8 Temps de calcul avec sélection d'information. 78

6.1 Localisation en ligne. 85
6.2 Schéma d'utilisation simultanée de plusieurs nœuds du graphe. 87
6.3 Résultats d'utilisation simultanée de plusieurs nœuds du graphe. 89
6.4 Localisation autour d'une image sphérique. 92
6.5 Localisation dans un graphe d'images sphériques. 93
6.6 Localisation dans un graphe d'images perspectives. 94
6.7 Localisation en environnement intérieur. 96
6.8 Trajectoire en environnement intérieur. 97

7.1 Position des images dans la scène. 103
7.2 Exemple de recalage hybride. 105
7.3 Histogrammes des erreurs MB et VO. 106
7.4 Vitesse de convergence des techniques MB,VO et H. 107
7.5 Mise en évidence de la dérive de l'odométrie visuelle. 108
7.6 Suivis avec changement d'illumination A. 110
7.7 Suivis avec changement d'illumination B. 111
7.8 Suivis avec changement d'illumination C. 112

8.1 Véhicule Cycab et ses capteurs. 114
8.2 Système de navigation autonome. 115
8.3 Modèle cinématique du véhicule. 116
8.4 Erreur régulée lors du suivi de trajectoire. 117
8.5 Trajectoire suivie à Sophia Antipolis. 119
8.6 Erreurs de pose à Sophia Antipolis. 119
8.7 Trajectoire suivie à Clermont-Ferrand. 120
8.8 Navigation autonome à Clermont-Ferrand 121
8.9 Exemple de façades à "l'infini". 122
8.10 Exemple d'ombres portées. 122
11 Gradients photométriques, ... 135
12 Interface DECSlam. 136
13 DECSlam. 137

Liste des algorithmes

2.1 Inverse compositionnelle 29
2.2 ESM 31
2.3 Suivi multi-résolution 34
5.1 Algorithme de sélection 75
7.1 Algorithme de suivi hybride. 106

Introduction et notations

Introduction

Depuis plusieurs années, la conception de véhicules terrestres assistés ou autonomes est un axe de recherche particulièrement actif dans le domaine de la robotique mobile. En ce qui concerne l'assistance à la conduite, des systèmes fiables sont déjà commercialisés sur des véhicules de série, tel que l'assistance au parking, la détection des panneaux de signalisation et de franchissement de ligne blanche, la correction de trajectoire ou le freinage d'urgence en cas de collision imminente.

Pour ce qui est des véhicules complètement autonomes, le champ d'application peut être l'exploration de zones dangereuses ou difficilement accessibles aux humains (*e.g.* mines, exploration spatiale ou sous marine *etc*), mais aussi la sécurisation des transports et l'optimisation du trafic routier. En effet, le réseau routier actuel est en limite de saturation en milieu urbain, une automatisation des véhicules est une solution contre l'engorgement des zones urbaines et pour une meilleure sécurité des transports.

Dans cette optique, plusieurs travaux ont déjà montré des résultats convaincants sur la faisabilité d'un véhicule autonome. Le *DARPA Grand Challenge* (Defense Advanced Research Projects Agency), est une compétition qui a été organisée dans le but d'accélérer le développement des véhicules automatiques, et qui a montré des résultats de navigation autonome sur de grandes distances dans le désert, et en milieu urbain. Plusieurs projets de recherche Européens (CyberCars, CyberMove) ou Français (Bodega,MobiVIP) ont également étudié le problème de navigation autonome en environnement urbain. Très récemment, *Sebastian Thrun et Chris Urmson* associés à l'entreprise Google ont présenté leurs résultats de navigation automatique (*cf.* Guizzo (2011)). Leur flotte de véhicules a parcouru un total de 300 000 kilomètres dans le flux de circulation, en centre ville et sur autoroute de manière autonome, avec un minimum d'intervention humaine. Alors que l'on pourrait penser le problème résolu, il reste en effet un travail important, notamment sur l'amélioration de la robustesse et de la fiabilité des techniques employées. De plus, pour obtenir ces résultats, *Thrun et Urmson* utilisent principalement des capteurs laser haut de gamme, dont la commercialisation à grande échelle n'est pas encore envisageable. Il est nécessaire de développer des méthodes alternatives basées sur des capteurs bas coût : les systèmes de vision sont une bonne option, car ils fournissent une information très riche sur l'environnement, qui peut être utilisée pour la navigation.

Une étape fondamentale pour la navigation autonome d'un robot est la localisation. Cette action anodine est effectuée en permanence par les êtres vivants pour se déplacer dans l'espace. Différents mécanismes sensoriels sont utilisés, en particulier la vision, mais aussi l'ouïe, l'odorat, le champ magnétique terrestre pour certains oiseaux migrateurs, un système sonar pour les chauve-souris ou les dauphins, ou encore un compas solaire pour certaines fourmis. Ces systèmes de localisation reposent sur des mécanismes naturels complexes, étudiés en particulier par les biologistes tels que Dyer (1996); Wehner *et al.* (1996) pour la

navigation des insectes, ou des psychologues comme Marr *et al.* (2010) pour la perception visuelle humaine. Ces mécanismes ne sont cependant pas encore compris parfaitement. En robotique mobile, l'idée principale est de reproduire ces mécanismes, en utilisant des capteurs spécifiques, afin d'effectuer une tâche de manière autonome. L'étape de localisation est cruciale : en effet avant de planifier un déplacement, il est préférable de connaitre sa position par rapport aux obstacles, de savoir si le chemin est navigable, ou simplement de connaitre la position du véhicule pour naviguer vers la destination désirée. La mémoire visuelle joue également un rôle très important dans la localisation : il est bien plus facile de se localiser lorsqu'un lieu a déjà été visité, ou lorsque l'on possède des données pouvant aider à la localisation (*e.g.* plan).

Contexte de la thèse

Cette thèse a été effectuée entre décembre 2008 et février 2012, à l'INRIA Sophia Antipolis dans le cadre du projet ANR predit CityVIP [1], sous la direction de Patrick Rives, directeur de recherche de l'équipe INRIA AroLag (ex. Arobas) et de Andrew Comport, chargé de recherche au laboratoire CNRS-I3S. Dans la continuité du projet MobiVIP et du développement de transports individuels autonomes, ce projet a pour but de développer des solutions de navigation autonome en environnement urbain.

Objectifs

L'objectif de ce travail est de définir une méthode de localisation en environnement urbain basée sur un capteur bas coût : une caméra. La méthode doit fournir un positionnement précis de la caméra, en utilisant une carte de l'environnement construite lors d'une phase d'apprentissage, afin de garantir l'intégrité du processus de navigation/localisation d'un véhicule autonome. Pour construire cette carte, il est nécessaire d'acquérir et de traiter une importante quantité de données lors de la phase d'apprentissage. Le premier problème consiste à définir de quelle manière ces données doivent être représentées, pour garantir une utilisation efficace de la base de données lors de la localisation en ligne. Le deuxième problème consiste à définir une méthode de localisation visuelle permettant le positionnement précis et temps-réel d'une caméra à l'aide de la base de données. La représentation des données et la technique de localisation visuelle utilisée sont fortement interdépendantes. En effet certaines représentations ne sont pas adaptées à certaines méthodes de localisation : il faut donc choisir une représentation des données permettant d'exploiter au maximum les avantages de la technique de localisation visuelle utilisée.

Contributions

Les travaux effectués durant cette thèse ont donné lieu à plusieurs publications scientifiques, directement liées à la localisation en environnement urbain et à la navigation autonome, mais aussi sur des problématiques plus générales de suivi visuel robuste ou de cartographie et localisation simultanée temps-réel :

1. Dans Meilland *et al.* (2010a,b), une nouvelle représentation sphérique égo-centrée a été présentée, ainsi qu'une nouvelle méthode de sélection de pixels ne dégradant pas l'observabilité des mouvements 3D.

[1] http ://projet_cityvip.byethost33.com

2. Dans Meilland *et al.* (2011a), un nouveau capteur permettant l'acquisition d'images sphériques augmentées par la profondeur est proposé. Ce capteur est utilisé pour cartographier automatiquement des environnements à grande échelle. Le modèle obtenu est alors utilisé pour localiser en temps réel une caméra navigant dans le voisinage du modèle.
3. Dans Meilland *et al.* (2011b), une méthode directe de suivi visuel temps réel, robuste aux changements d'illumination est proposée. Cette méthode se base sur l'utilisation d'une technique de localisation basée modèle et d'une technique basée odométrie visuelle.
4. Dans Gallegos *et al.* (2010), le modèle de sphères augmentées est utilisé pour une application de cartographie et localisation simultanée basée sur une fusion des données d'une caméra omnidirectionnelle et d'un laser 2D.
5. Dans Audras *et al.* (2011), les algorithmes développés durant cette thèse sont utilisés pour un système de cartographie et localisation simultanée temps-réel basé sur une caméra RGB+D.
6. Dans Comport *et al.* (2011), un système de cartographie stéréo, puis de localisation monoculaire temps-réel, a été proposé. Cette publication a donné lieu à des démonstrations temps-réel lors de la conférence.
7. Les algorithmes de cartographie et de localisation temps réels développés durant cette thèse, ont été rassemblés dans une librairie nommée *DECSlam* (*Dense ego-centric simultanemous localisation and mapping*), et déposés dans une APP INRIA (Agence pour la protection des programmes *cf.* Annexe V).

Organisation du manuscrit

Ce manuscrit est divisé en trois principales parties :

1. Dans la première partie, les différentes méthodes de localisation visuelle de la littérature sont détaillées dans le premier chapitre. Cet état de l'art permet de définir la problématique générale de cette thèse, et permet également de justifier les choix effectués pour la suite du manuscrit. Le deuxième chapitre définit la partie théorique du suivi visuel direct utilisé.
2. La deuxième partie définit la phase hors-ligne, c'est à dire l'apprentissage : une cartographie de l'environnement est présentée sous la forme d'un graphe d'images sphériques augmentées par une information dense de profondeur. Dans cette partie, plusieurs contributions sont proposées :
   - Une nouvelle représentation sphérique égo-centrée, permettant de cartographier de manière dense des environnements de grande échelle.
   - Une nouveau capteur sphérique, permettant d'acquérir des images panoramiques augmentées par l'information de profondeur dense.
   - Une nouvelle méthode de sélection de pixels saillants maximisant l'observabilité des mouvements 3D locaux.
3. La troisième partie définit la phase en ligne : la base de données construite hors-ligne est utilisée pour la localisation efficace d'une caméra et la navigation autonome d'un véhicule en environnement urbain. Cette partie introduit différentes contributions :

- Une méthode de localisation visuelle directe exploitant les avantages du graphe d'images augmentées pour une localisation temps-réelle, précise et robuste, fonctionnant en intérieur et en extérieur.
- Une nouvelle méthode de suivi visuel 3D hybride, robuste aux changements d'illumination et aux occultations, couplant localisation basée modèle et odométrie visuelle.

Notations

Règles générales

a	Scalaire
$\mathbf{a}$	Vecteur
$\mathbf{A}$	Matrice

Mathématiques

$\mathbf{A}^T$	Transposée de $\mathbf{A}$
$\mathbf{K}$	Matrice des paramètres intrinsèques
$\mathbf{T}$	Matrice de pose
$\mathbf{R}$	Matrice de rotation
$\mathbf{t}$	Vecteur de translation
$\mathbf{x}$	Vecteur de pose inconnu
$\tilde{\mathbf{x}}$	Vecteur de pose réel
$\widehat{\mathbf{x}}$	Vecteur de pose estimé
$\mathbf{M}$	Matrice de projection perspective
$\mathbf{P}$	Point 3D
$\overline{\mathbf{P}}$	Point 3D en coordonnées homogènes
$\mathcal{I}$	Image
$\mathbf{p}$	Coordonées 2D d'un pixel
$\overline{\mathbf{p}}$	Coordonées 2D homogènes d'un pixel
$\mathbf{e}$	Vecteur d'erreur
$\overline{\mathbf{e}}$	Vecteur d'erreur centré
$\mathbf{J}$	Matrice jacobienne
$[\cdot]_\times$	Matrice antisymétrique
$w(\cdot)$	Fonction de *warping*
$\rho(\cdot)$	Fonction robuste

Acronymes

SLAM	*Simultaneous Localisation And Mapping*
GPS	*Global Positionning System*
CAO	*Conception Assistée par Ordinateur*
SfM	*Structure from Motion*
IC	*Inverse Compositionnelle*
ESM	*Efficient Second Order Minimisation*
IRLS	*Iterative Re-weighted Least Squares*
SSD	*Sum of Squared Differences*
ZNCC	*Zero Normalized Cross Correlation*
MAD	*Median Absolute Deviation*
VO	*Visual Odometry*
MB	*Model Based*
SIFT	*Scale Invariant Feature Transform*

Première partie

Localisation

Chapitre 1

État de l'art

1.1 Introduction

Pour localiser un véhicule, le capteur par excellence est le système *GPS* (*Global Positionning System*), car il fournit une position géo-référencée du récepteur en mesurant le temps de réception des signaux émis par des satellites. Ce genre de capteur, aujourd'hui très répandu, est employé dans tous les types de transport, terrestres, maritimes, aériens, et est également intégré sur de nombreux téléphones mobiles. Cependant, l'erreur de précision d'un récepteur bas coût est de l'ordre de quelques dizaines de mètres, ce qui est bien trop important pour localiser un véhicule autonome et planifier des trajectoires sur des espaces navigables : une voie de circulation mesure en moyenne 3.50 mètres de large.

De plus, en environnement urbain, les bâtiments peuvent occulter les satellites, notamment lorsque le véhicule se trouve dans un environnement de type *canyon urbain*. Sachant qu'il faut un minimum de 4 satellites pour une localisation correcte, il est fréquent que le nombre de satellites visibles ne soit pas suffisant pour se localiser. A cela s'ajoute le phénomène de *multi-trajet* : les signaux *GPS* peuvent être réfléchis sur les façades des bâtiments, ce qui peut fausser la mesure effectuée. De plus un système *GPS* classique ne fournit pas directement une information d'orientation : elle doit être déduite du mouvement ou calculée en utilisant deux récepteurs suffisamment espacés.

Pour se localiser dans les régions sans réception *GPS*, certains systèmes utilisent des capteurs proprioceptifs, qui mesurent une information sur le mouvement du robot :

- Encodeurs placés sur les roues d'un véhicule : mesure de vitesses.
- Centrale inertielle : mesure d'accélérations, vitesses angulaires et cap.

Ces capteurs fournissent des mesures de vitesses et d'accélérations qu'il faut intégrer une ou deux fois pour obtenir un déplacement, ce qui entraine des erreurs relativement importantes sur de longues distances (dérive). De plus les capteurs de type odomètre fournissent un mouvement uniquement dans le plan de la route, et produisent des mesures aberrantes si les roues du véhicule glissent sur le sol. Bien qu'ils offrent une bonne complémentarité aux autres capteurs, en particulier à cause de leur fréquence d'acquisition (*i.e.* 100 Hz), il n'est pas envisageable d'utiliser uniquement des capteurs proprioceptifs pour la localisation.

Pour améliorer la qualité de la localisation, il est possible d'utiliser des capteurs extéroceptifs, qui donnent une mesure sur l'environnement (*i.e.* géométrie, photométrie). Un capteur de type télémètre laser permet de mesurer une distance : en général une coupe 2D de la géométrie de l'environnement, qui peut être utilisée pour estimer le mouvement

du robot dans le plan du laser avec des techniques telles que le *Scan-Matching* Diosi & Kleeman (2007) ou l'*Iterative Closest Point* Zhang (1994). Certains capteurs très haut de gamme permettent d'obtenir un nuage 3D en utilisant plusieurs nappes laser superposées, ou un capteur 2D rotatif. Cependant ces dispositifs sont lourds à utiliser en temps réel à cause de la quantité de mesures effectuées, et sont surtout particulièrement onéreux.

Alternativement, les caméras sont des capteurs extéroceptifs bas coût, fournissant une information très riche, précise, utilisable pour la localisation. C'est ce type de capteur qui va être utilisé dans la suite de ce manuscrit. Contrairement aux capteurs laser, une image fournit une information projective de l'environnement, pour obtenir une information de localisation ou de géométrie, il est nécessaire de traiter les images. La suite, de ce chapitre présente les différentes méthodes de localisation visuelle, ainsi que les choix effectués pour les algorithmes de localisation développés dans cette thèse.

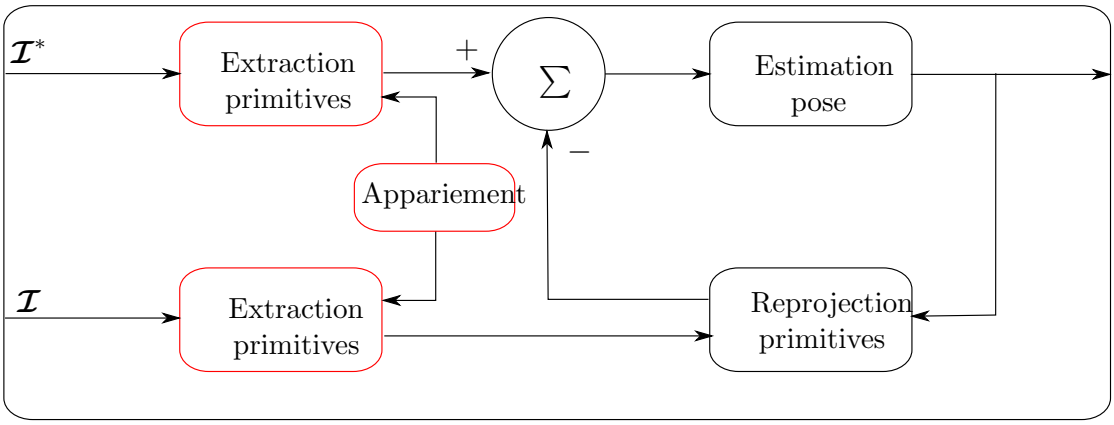
1.2 Estimation du mouvement d'une caméra

La localisation visuelle peut être considérée comme l'estimation de la trajectoire d'une caméra par rapport à un repère initial. Ce repère peut être par exemple la première image d'une séquence, ou alors un objet connu, suivi dans les images et permettant d'extraire une information sur la position relative entre la caméra et l'objet. Tout d'abord, les techniques de localisation visuelle peuvent être divisées en deux groupes : les méthodes basées points d'intérêts et les méthodes directes. Ensuite le problème de localisation est étroitement lié à l'estimation de la géométrie de la scène, on parle alors de localisation et cartographie simultanée (*SLAM* : *Simultaneous Localisation and Mapping*). Enfin, pour simplifier le problème de localisation, il est possible d'effectuer la partie cartographie lors d'une phase d'apprentissage et d'utiliser le modèle appris pour localiser une caméra.

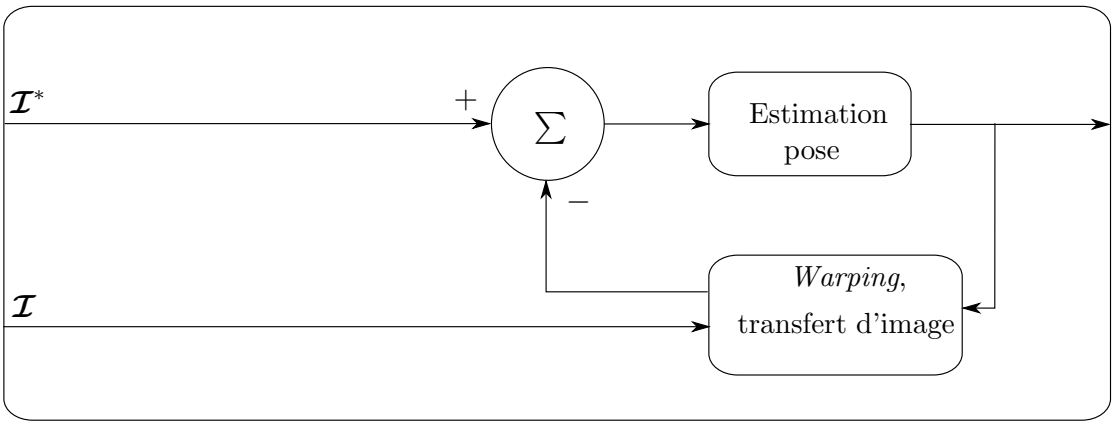
1.2.1 Méthodes basées points d'intérêt

Les méthodes basées sur l'extraction de points d'intérêts (*feature-based*), cherchent à extraire des primitives visuelles locales dans les images, telles que les points de Harris & Stephens (1988). Une étape d'appariement est ensuite effectuée, c'est à dire mettre en correspondance un à un les points d'intérêts de chaque image. Pour cela il est nécessaire d'utiliser une mesure de similarité entre les primitives. En général le coût est évalué entre les intensités des pixels extraits dans voisinage des points d'intérêts, par une *SSD* (*Sum of Squared Differences*), une *SAD* (*Sum of absolute differences*) ou une *ZNCC* (*Zero Normalized Cross Correlation*). Pour rendre plus robuste cette étape, souvent critique pour ce genre de méthodes, il est possible d'utiliser des descripteurs locaux, robustes aux changements d'echelle et aux rotations, tels que Lowe (2004); Bay *et al.* (2006). Le mouvement relatif entre les images est en général obtenu soit par une méthode robuste de type *RANSAC* (*R*andom Sample Consensus Fischler & Bolles (1981)), soit par une méthode itérative de minimisation non linéaire de l'erreur de re-projection des points d'intérêts (*cf.* figure 1.1(a)).

Ce genre d'approches, sont les plus fréquemment utilisées pour l'estimation de pose entre images, car elles permettent de réduire l'information contenue dans les images à quelques centaines de points d'intérêts. Cependant elles nécessitent une étape intermédiaire d'extraction et d'appariement de points d'intérêts entre les images, basée en général sur des seuils de détection. Cette étape de pré-traitement est souvent mal conditionnée, bruitée et non robuste, ce qui nécessite d'utiliser des techniques d'estimation robustes de haut niveau.



(a) Méthode basé points d'intérêts.



(b) Méthode directe.

FIG. 1.1 – Comparaison des méthodes d'estimation itératives. (a) Une méthode basée points d'intérêts nécessite l'extraction de primitives dans les images, puis une étape d'appariement. La pose entre les images peut alors être estimé itérativement en minimisant la re-projection des points d'intérêts. (b) Une méthode directe minimise directement une erreur d'intensité entre les deux images, en utilisant une technique de synthèse d'image.

1.2.2 Méthodes directes

Les méthodes directes (*image-based*), quant à elles, n'ont pas de phase de sélection de points d'intérêts ou de primitives visuelles. Le mouvement de la caméra est directement obtenu en minimisant les erreurs d'intensités communes aux deux images (*cf.* figure 1.1(b)) à l'aide d'une transformation paramétrique. Dans ce cas l'estimation du mouvement et la mise correspondance des pixels s'effectuent simultanément lors de l'optimisation. Dans la majorité des cas, ce type de technique est utilisé pour le suivi d'une surface planaire Lucas & Kanade (1981); Irani & Anandan (1998, 2000); Baker & Matthews (2001); Malis (2004); Dame & Marchand (2010), ou alors le suivi d'un ensemble de surfaces planaires Mei *et al.* (2006); Silveira *et al.* (2008); Shi & Tomasi (1994); Caron *et al.* (2011). Quelques travaux, notamment ceux présentés dans Comport *et al.* (2007) proposent une généralisation de l'algorithme au suivi de modèles 3D obtenus par mise en correspondance dense stéréo. Tous les pixels des images sont alors utilisés dans la boucle d'estimation, permettant une estimation robuste et précise du mouvement.

1.3 *SLAM*

Le problème de localisation et de cartographie simultanée (*SLAM*) a été ces dernières années un axe de recherche particulièrement actif dans le domaine de la robotique (*cf.* Montemerlo *et al.* (2002); Davison & Murray (2002); Thrun (2002); Durrant-Whyte & Bailey (2006); Klein & Murray (2007); Konolige & Agrawal (2008)). A partir d'une position de départ, le principe consiste à construire incrémentalement une carte de l'environnent, d'utiliser cette carte pour la localisation et de mettre la carte à jour lorsque de nouvelles mesures sont effectuées. Grâce à l'ajout de nouvelles données, l'incertitude sur la carte diminue. Classiquement, les techniques de *SLAM* visuel sont basées sur des points d'intérêts et un filtre de Kalman étendu Jazwinski (1970). Cependant ces méthodes ont une efficacité calculatoire limitée, due à l'inversion d'une matrice de covariance dont les dimensions augmentent avec la taille de la carte reconstruite. De plus, ce genre d'approches est sujet à des problèmes de consistance liés à la linéarisation des modèles. Pour conserver des solutions temps réel, le *SLAM* peut être effectué sur des fenêtres de visibilité, tel que dans Mouragnon *et al.* (2006); Mei *et al.* (2010). Lorsque, seulement la trajectoire de la caméra est considérée, on parle alors d'odométrie visuelle Nistér *et al.* (2004); Howard (2008); Kitt *et al.* (2010); Tardif *et al.* (2010); Comport *et al.* (2010) qui consiste à estimer incrémentalement la position de la caméra, le long de la trajectoire.

Récemment, des techniques de *SLAM* dense basées sur des méthodes directes ont été proposées. Tykkala & Comport (2011) proposent d'intégrer temporellement les cartes de disparités obtenues par mise en correspondance stéréo afin d'améliorer la reconstruction, en réduisant la zone de recherche des disparités. Dans Newcombe *et al.* (2011b), un système de *SLAM* monoculaire direct est proposé pour reconstruire des modèles denses d'environnements intérieurs de dimensions limitées.

Cependant, toutes les approches présentées précédemment restent incrémentales, et intègrent une erreur de dérive, non négligeable sur de longues distances. Certains algorithmes proposent de détecter les lieux ou le robot est déjà passé, en fonction de l'apparence des images, on parle alors de fermeture de boucle (*cf.* Cummins & Newman (2008); Williams *et al.* (2009); Chapoulie *et al.* (2011)). Ce genre d'information permet d'ajouter des contraintes sur la trajectoire obtenue, qui peut être ré-optimisée efficacement par certains algorithmes dédiés à l'optimisation de graphes tels que Grisetti *et al.* (2007); Konolige (2010); Kummerle *et al.* (2011).

Dans le domaine de la vision par ordinateur, le problème de *SLAM* est plus connu sous le nom de *Structure from Motion*(*SfM*) (*cf.* Martinec *et al.* (2002); Seitz *et al.* (2006); Furukawa & Ponce (2010)). Bien que l'objectif soit similaire, c'est à dire reconstruire le mouvement des caméras et la structure de la scène, les contraintes sont différentes. En effet, contrairement au *SLAM* où la reconstruction est effectuée de manière incrémentale et en ligne, toutes les images nécessaires à la reconstruction sont disponibles à l'initialisation. En général, les reconstructions sont effectuées hors-ligne, ce qui permet d'utiliser des techniques d'ajustement de faisceaux (Triggs *et al.* (2000)), consistant à optimiser simultanément toutes les variables du système (*i.e.* position des images et modèle 3D). Bien que certains algorithmes fournissent des reconstructions sur des régions de taille relativement importante (Furukawa & Ponce (2010)), la reconstruction réaliste à très grande échelle reste pratiquement irréalisable.

Lorsque qu'une seule caméra est utilisée pour le *SLAM* visuel, on parle de *SLAM* mo-

noculaire ou *Bearing Only SLAM*. Puisque seulement un capteur projectif est utilisé, il faut au minimum deux observations à des positions différentes pour contraindre la reconstruction d'un amer. De plus la localisation de la caméra et la reconstruction de la carte sont obtenus à un facteur d'échelle près. Dans certaines applications, notamment en réalité augmentée ce facteur n'est pas toujours important. Cependant pour une application de navigation autonome il est dans certains cas indispensable d'avoir une localisation à l'échelle, par exemple pour envoyer des commandes cohérentes au robot. Pour corriger ce facteur, il est possible d'utiliser des capteurs proprioceptifs tel que dans Royer *et al.* (2005). Une seconde solution est d'utiliser un système de stéréo vision, c'est à dire deux caméras reliées rigidement entre elles, dont la position est parfaitement connue (obtenue lors d'une phase d'étalonnage). Ces systèmes facilitent le problème de *SLAM*, car l'observation de l'information 3D est possible sans devoir estimer simultanément la localisation, et sans nécessiter de déplacements spéciaux de la caméra pour assurer l'observabilité. D'autres approches associent un capteur extéroceptif de type télémètre laser à une caméra tel que Gallegos *et al.* (2010), ce qui permet d'obtenir directement une information métrique de profondeur dans les images.

1.4 Méthodes avec apprentissage

Un algorithme de *SLAM* en temps réel n'étant pas envisageable à grande échelle pour la localisation d'un robot, il est possible de découpler le problème en deux parties :

1. La cartographie, la partie la plus complexe, peut être traitée hors-ligne lors d'une phase d'apprentissage.
2. La carte obtenue peut alors être utilisée en ligne, pour localiser efficacement une caméra naviguant à l'intérieur du modèle.

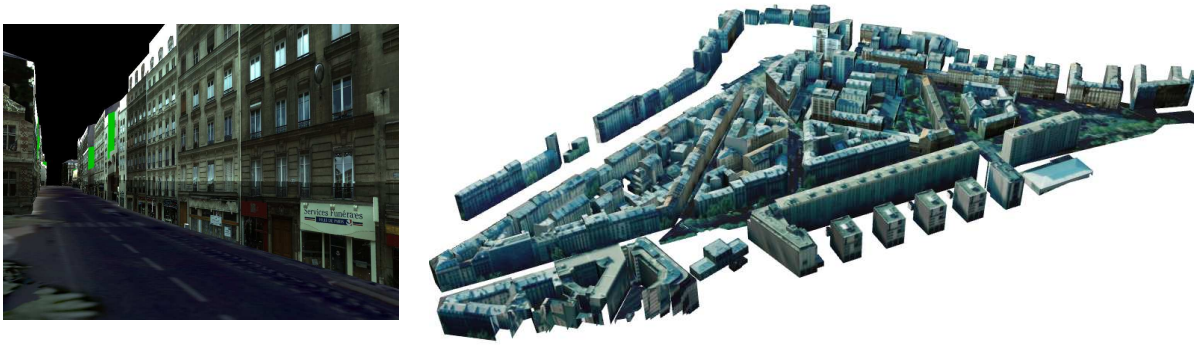
Ce genre d'approche a plusieurs avantages, d'une part la localisation peut être effectuée avec précision et sans dérive, grâce au modèle. De plus si le modèle 3D est à l'échelle, la localisation visuelle peut être effectuée avec une caméra monoculaire, à l'échelle également.

Ce type de méthodes, peut également être classé en deux groupes : l'utilisation de modèles 3D, obtenus soit par conception assisté par ordinateur (*CAO*), soit avec une méthode de reconstruction automatique, et les modèles de type "mémoires images", consistant à distribuer dans l'environnement des images acquises lors de la phase d'apprentissage, sans reconstruire explicitement le modèle 3D global.

1.4.1 Modèles 3D

Certains algorithmes exploitent directement un modèle *CAO* de l'objet à suivre (*cf.* Brown (1971); Lowe (1991); Marchand *et al.* (2001); Drummond *et al.* (2002); Vacchetti *et al.* (2004); Comport (2005); Comport *et al.* (2006)). La position de la caméra est estimée par rapport à l'objet en minimisant l'erreur de re-projection entre le modèle 3D de la cible et les contours extraits dans les images. Cependant ces algorithmes nécessitent une bonne modélisation des objets ainsi que des primitives visuelles structurées dans les images, telles que des droites pour fonctionner.

Plusieurs travaux ont étés menés pour améliorer les techniques de localisation en environnement urbains en utilisant un modèle *CAO*. Lothe *et al.* (2010) utilisent un modèle 3D global approximatif, pour recaler en ligne une carte locale reconstruite par un algorithme de *SLAM* visuel avec la partie géométrique du modèle, afin de corriger la dérive. Dans Cappelle



(a) Image synthétique. (b) Modèle 3D texturé.

FIG. 1.2 – (a). Une image synthétisée à partir du modèle 3D texturé (b). *Source : Institut Géographique National (IGN).*

et al. (2011), le modèle 3D est seulement utilisé pour détecter les obstacles entre les images perçues par une caméra et les images virtuelles, la localisation de la caméra étant obtenue par *GPS*-RTK. Dans Irschara *et al.* (2009), un modèle éparse de points 3D reconstruit par un algorithme de *SfM* est utilisé pour localiser une caméra par un appariement de points *SIFT*.

En général, ces modèles représentent d'une manière approximative l'environnement ou les objets à suivre dans les images. Bien que les méthodes de reconstruction automatique d'environnement urbains à grande échelle deviennent de plus en plus précises Hammoudi *et al.* (2010); Craciun *et al.* (2010); Lafarge & Mallet (2011), les outils utilisés et les modèles reconstruits sont principalement dédiés à des applications de réalité virtuelle (*cf.* figure 1.2). En effet ce genre de modèle, obtenu par plaquage de textures sur un bâtit 3D approximatif (façades planaires), ne permet pas un rendu photo-réaliste de l'environnement et comporte des erreurs de modélisation et des inconsistances photométriques. Pour être robuste à ces erreurs, Caron *et al.* (2012) proposent d'utiliser l'information mutuelle (Viola & Wells (1995)) pour recaler une image de synthèse, générée à partir d'un modèle 3D texturé avec une image réelle. Cette métrique permet de traiter des images de modalités différentes, cependant les calculs nécessaires pour l'alignement ne sont pas temps-réel.

Dans Newcombe *et al.* (2011b), le modèle 3D dense obtenu par un algorithme de *SLAM* est ré-utilisé pour localiser une caméra avec une méthode directe, pour une application de réalité augmentée. La pose de la caméra est estimée en minimisant directement les intensités de l'image courante, avec celles de l'image virtuelle. Bien que le modèle soit quasiment photo-réaliste, l'espace de reconstruction est restreint à un environnement réduit (*e.g.* bureau).

1.4.2 Mémoires image

Une approche alternative à la reconstruction d'un modèle 3D global, consiste à représenter l'environnement de manière égo-centrée : les approches basées mémoire image proposent de conserver dans la base de donnée directement les images issues de la phase d'apprentissage, positionnées en 2 ou en 3 dimensions dans l'espace. L'information locale contenue dans les images de la base de donnée est alors utilisée en ligne pour re-localiser une caméra naviguant dans le voisinage de la base de données. Contrairement aux méthodes globales, ces approches fournissent localement un maximum de précision. En effet, les données extraites des images ne sont pas exprimées dans un repère global, ce qui évite de propager les erreurs liées à la reconstruction et aux approximations géométriques des modèles, ce qui améliore la précision de la localisation.

Dans Royer *et al.* (2005) une base données d'images clés, contenants des points de Harris ainsi que leur position 3D, est construite lors d'une phase d'apprentissage. La base de données est alors utilisée pour localiser en ligne une caméra monoculaire en utilisant une méthode basée points d'intérêts. Courbon *et al.* (2009) proposent une méthode basée sur un graphe d'images générique (image fisheye, omnidirectionnelle) définissant un chemin visuel à suivre. Le graphe est utilisé pour du suivi de trajectoire. Cobzas *et al.* (2003) construisent une base de données d'images panoramiques augmentées par la profondeur avec un laser et une caméra montés sur une tourelle pan/tilt. Une méthode basée point d'intérêts permet de re-localiser une caméra monoculaire en environnement intérieur. Menegatti *et al.* (2004) proposent une base de donnée d'images omnidirectionnelles pour une localisation qualitative en intérieur. Dans Jogan & Leonardis (2000) la méthode de localisation proposée est basée sur des panoramas cylindriques. Zhang & Kosecka (2006) utilisent une mémoire image positionnée avec un système GPS, pour de la reconnaissance de lieux et une localisation visuelle basée points d'intérêts.

D'autres approches utilisent une mémoire visuelle pour effectuer de l'asservissement visuel Mezouar & Chaumette (2003); Remazeilles *et al.* (2004); Segvic *et al.* (2007); Dame & Marchand (2011). Ce type d'approches ne calculent pas explicitement une position 3D, le contrôle effectué par l'asservissement est directement calculé en fonction d'une erreur de re-projection dans les images. Ce genre de techniques ne permettent pas de générer des trajectoires différentes de celles de l'apprentissage, ce qui peut être nécessaire dans le cas d'une application de navigation en environnement urbain (*e.g.* obstacles, dépassement *etc*). Cependant, Cherubini & Chaumette (2011) ont proposé de déformer localement la trajectoire pour effectuer de l'évitement d'obstacles en utilisant une tourelle pan/tilt pour conserver les images de la base de données dans le champ de vue de la caméra.

1.5 Conclusion

Les techniques de localisation visuelle basées points d'intérêts, nécessitent une étape intermédiaire d'extraction et d'appariement de points entre les images, basée en général sur des seuils de détection. Cette étape de pré-traitement est souvent mal conditionnée, bruitée et non robuste, ce qui nécessite d'utiliser des techniques d'estimation robustes de haut niveau. Dans ces travaux, il a été choisi d'utiliser une méthode de localisation visuelle directe dérivée de Comport *et al.* (2007, 2010), minimisant itérativement une erreur d'intensité entre tous les pixels des images. Ce genre d'approche permet d'estimer avec précision le mouvement 3D d'une caméra, tout en étant robuste aux erreurs de modélisation grâce à

l'utilisation de toute l'information contenue dans l'image.

Dans le cadre d'une application de navigation autonome en environnement urbain, il est tout à fait possible de supposer que l'environnement puisse être cartographié lors d'un apprentissage, et la carte utilisée pour localiser un véhicule en 3D, sans nécessiter de reconstruction coûteuse en ligne. Dans ce cas l'objectif est d'effectuer un maximum de calculs hors ligne, pour bien conditionner la phase de localisation en ligne. Cependant, comparer l'image virtuelle d'un modèle 3D global contenant une géométrie approximative, avec l'image réelle perçue par une caméra, n'est pas la meilleure approche, en particulier pour une méthode directe minimisant directement une erreur d'intensité. De plus les modèles 3D existants sont consistants globalement, mais n'ont aucune précision localement. En revanche, une carte d'apprentissage égo-centrée basée sur une mémoire image, fournit des données directement issues de capteurs photométriques, permettant de minimiser une erreur consistante entre les images de la base de donnée et l'image courante perçue par une caméra, ce qui est bien adapté à l'utilisation de méthodes directes. De plus, le fait d'utiliser des données géométriques directement mesurées dans les images évite de propager les erreurs de reconstruction globales ce qui permet une localisation précise, nécessaire à la navigation.

Finalement, il est possible de faire le lien entre les méthodes de localisation égo-centrées et les techniques de rendu graphique basées image (*Image-based rendering* Gortler *et al.* (1996); Levoy & Hanrahan (1996); Debevec *et al.* (1996)), qui utilisent directement les images originales utilisées lors la reconstruction géométrique, sans effectuer un plaquage de texture sur le modèle 3D global. Les images synthétisées avec ces méthodes sont photo-réalistes car elles conservent la finesse et la précision des images originales, ce qui est similaire à l'approche égo-centrée développée dans cette thèse.

Chapitre 2

Suivi visuel 3D direct

2.1 Introduction

Ce chapitre détaille les bases théoriques de la technique de suivi visuel 3D direct, utilisée dans ces travaux. La première partie définit tout d'abord quelques notions de géométrie et de transfert d'image. La deuxième partie définit une fonction de coût, reliant la transformation rigide entre deux capteurs par une erreur d'intensité directement calculée entre deux images. Cette fonction de coût peut être minimisée par une technique d'optimisation non linéaire robuste, permettant d'estimer avec précision le mouvement 3D d'une caméra. La méthode décrite est dérivée de Comport *et al.* (2010), qui utilisent une transformation quadrifocale, exprimée en fonction d'une disparité dans les images pour synthétiser de nouvelles vues. Pour plus de généricité, la méthode développée formule le problème de transfert d'image directement avec une information de profondeur entre la caméra et la scène.

2.2 Notions de géométrie

2.2.1 Transformation rigide

Soit un repère orthonormé $\mathcal{F}^*$ appartenant à l'espace Euclidien, nommé repère de référence, et un repère $\mathcal{F}$ orthonormé nommé repère courant. Soit la matrice homogène $\mathbf{T} \in \mathbb{SE}(3) \subset \mathbb{R}^{4\times4}$, appartenant au groupe Spécial Euclidien, de dimensions 4×4 tel que :

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{2.1}$$

où $\mathbf{R} \in \mathbb{SO}(3) \subset \mathbb{R}^{3\times3}$ est une matrice de rotation, appartenant au groupe Spécial Orthogonal et $\mathbf{t} \in \mathbb{R}^3$ est un vecteur de translation de dimensions 3×1.

La matrice $\mathbf{T}$ définit le déplacement 3D rigide entre les repères $\mathcal{F}^*$ et $\mathcal{F}$, ou plus communément la transformation de pose entre les deux repères.

Soit $\mathbf{P}^* = \begin{bmatrix} X & Y & Z \end{bmatrix}^T \in \mathbb{R}^3$, un point 3D de l'espace Euclidien définit dans le repère $\mathcal{F}^*$. Le point $\mathbf{P}^*$ peut être transféré par la transformation rigide $\mathbf{T}$ dans le repère $\mathcal{F}$ par :

$$\overline{\mathbf{P}} = \mathbf{T}\overline{\mathbf{P}^*} \quad \text{ou} \quad \mathbf{P} = \mathbf{R}\mathbf{P}^* + \mathbf{t} \tag{2.2}$$

où $\overline{\mathbf{P}^*} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ correspond au coordonnées homogènes du point $\mathbf{P}^*$.

Les propriétés du groupe Spécial Orthogonal, permettent de définir les équations suivantes :

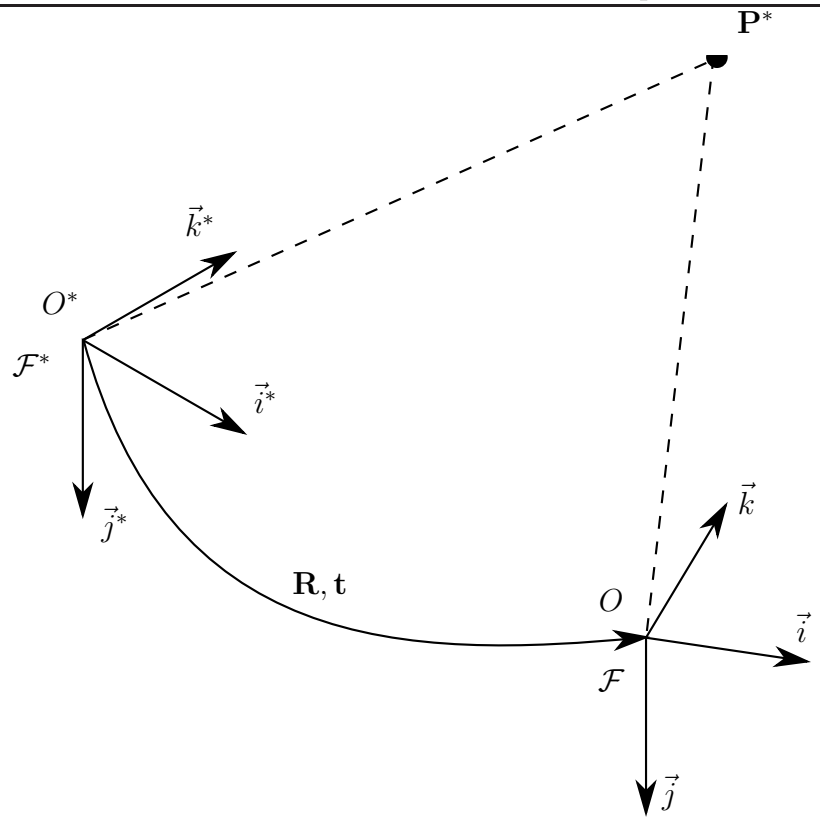


FIG. 2.1 – Transformation rigide entre les repères $\mathcal{F}^*$ et $\mathcal{F}$.

– L'inverse d'une matrice de rotation :

$$\mathbf{R}^T\mathbf{R} = \mathbf{I}. \tag{2.3}$$

– L'inverse d'une matrice de pose homogène :

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{2.4}$$

2.2.2 Transformation de vitesse

Soit un vecteur $\mathbf{x} \in \mathbb{R}^6$, représentant des vitesses instantanées en translation $\boldsymbol{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$ et en rotation $\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$, tel que :

$$\mathbf{x} = \int_0^1 (\boldsymbol{\omega}, \boldsymbol{v}) dt \in \mathfrak{se}(3), \tag{2.5}$$

Le vecteur $\mathbf{x}$ est relié à une pose $\mathbf{T}(\mathbf{x}) \in \mathbb{SE}(3)$ par l'application matrice exponentielle :

$$\mathbf{T}(\mathbf{x}) = \exp([\mathbf{x}]_\wedge) = \sum_{i=0}^{\infty} \frac{1}{i!}([\mathbf{x}]_\wedge) \tag{2.6}$$

où l'opérateur $[\cdot]_\wedge$ est défini par :

$$[\mathbf{x}]_\wedge = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \boldsymbol{v} \\ \mathbf{0} & 0 \end{bmatrix} \tag{2.7}$$

et l'opérateur $[\cdot]_\times \in \mathbb{SO}(3)$ défini la matrice antisymétrique du vecteur $\boldsymbol{\omega} = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^T$ tel que :

$$[\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.8}$$

2.3 Formation de l'image

2.3.1 Projection perspective

Une caméra perspective classique peut être modélisée par le modèle sténopé (*cf.* Faugeras (1993); Hartley & Zisserman (2004)). Tout d'abord, il est possible de définir la matrice des paramètres intrinsèques d'une caméra :

$$\mathbf{K} = \begin{bmatrix} f & s & u_0 \\ 0 & f \times r & v_0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3} \tag{2.9}$$

où f est la distance focale de la caméra exprimée en pixels, s le facteur de cisaillement, r est le rapport des dimensions d'un pixel, et le couple (u_0,v_0) correspond à la position du point principal en pixels (*i.e.* centre de l'image). En général, pour une caméra de bonne qualité, le facteur de cisaillement est nul : $s = 0$ et le rapport des dimensions est proche de 1 : $r \approx 1$.

Le point $\mathbf{P}^*$ se projette sur le plan image normalisé de la caméra par :

$$\overline{\mathbf{p}^*} = \frac{\mathbf{M}\overline{\mathbf{P}^*}}{\mathbf{e}_3^T\mathbf{M}\overline{\mathbf{P}^*}}, \tag{2.10}$$

où la matrice de projection perspective $\mathbf{M}$, de dimensions 3×4 est le produit matriciel de la pose et des paramètres intrinsèques :

$$\mathbf{M} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}, \tag{2.11}$$

et le vecteur $\mathbf{e}_3 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$ est un vecteur unitaire permettant d'extraire la troisième composante du produit $\mathbf{M}\overline{\mathbf{P}^*}$.

Le point projeté $\overline{\mathbf{p}^*} = \begin{bmatrix}u & v & 1\end{bmatrix}^T$ correspond aux coordonnées pixelliques normalisées dans l'image. Par convention, le point de coordonnées $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$ est associé au premier pixel en haut à gauche de l'image.

La matrice $\mathbf{K}$ peut être obtenue lors d'une phase d'étalonnage (*cf.* Tsai (1992); Heikkila & Silven (1997); Zhang (1999)), à partir d'images d'un objet ou d'une mire d'étalonnage dont les dimensions sont connues (échiquier, cercles ...). Pour un objectif à focale fixe, cette matrice est constante au cours du temps. Dans les chapitres suivants, pour une meilleure lisibilité, cette matrice sera considérée constante et parfaitement connue, elle n'apparaîtra que dans certaines équations.

2.3.2 Distorsions radiales

En général, le modèle sténopé n'est pas suffisant pour modéliser correctement la formation d'une image. Des aberrations ou distorsions, causées par les lentilles des objectifs des

caméras sont présentes. En effet l'approximation de Gauss, assumant une propagation rectiligne de la lumière n'est valable que pour de petits angles d'incidence. Sur les bords d'un objectif (correspondant aux bords d'une image), cette approximation n'est plus valable : les lignes droites de l'environnement apparaissent courbées dans les images. Ce phénomène est d'autant plus prononcé que l'angle d'ouverture de la caméra (inversement proportionnel à la distance focale) est important.

Afin de corriger ces aberrations, un modèle de distorsion radial peut être utilisé (Slama (1980)). Le déplacement des pixels dans les images peut être modélisé par une fonction polynomiale d'ordre n :

$$\overline{\mathbf{p}^{\mathbf{r}}} = \mathbf{K}(1 + k_2 r^2 + k_4 r^4 + \ldots + k_n r^n)\mathbf{K}^{-1}\overline{\mathbf{p}^*}, \tag{2.12}$$

avec $r = \|\mathbf{K}^{-1}\overline{\mathbf{p}^*}\|$ correspondant à la distance du pixel par rapport au centre de projection, k_i les coefficients de distorsion du polynôme et $\mathbf{p}^{\mathbf{r}}$ la nouvelle position des pixels $\mathbf{p}^*$.

Tout comme les paramètres intrinsèques, ces coefficients peuvent être obtenus lors de l'étalonnage de la caméra (*cf.* Heikkila & Silven (1997); Zhang (1999)) et sont en général constant au cours du temps. Puisque la correction de distorsion ne dépend que de la position des pixels dans les images (*cf.* équation (2.12)), il est possible de corriger les intensités des images lors d'une phase de pré-traitement. Dans la suite de ce manuscrit, les images utilisées seront considérées comme corrigées de toute distorsion.

2.4 Transformation d'une image

2.4.1 Fonction de *warping*

Soit une image $\boldsymbol{\mathcal{I}}^*$ de dimensions $m \times n$ associée à une fonction d'intensité $\boldsymbol{\mathcal{I}}^*(\mathbf{p}^*)$ et à un repère $\mathcal{F}^*$. Les valeurs $\mathbf{p}^* = (u, v)$, avec $u \in [0; m[$ et $v \in [0; n[$, correspondent aux coordonnées des pixels de l'image $\boldsymbol{\mathcal{I}}^*$. Supposons ensuite que pour chaque pixel $\mathbf{p}^*$ une information de distance métrique $Z \in \mathbb{R}^+$ est connue. Un point 3D dans l'espace Euclidien est par conséquent défini par $\mathbf{P} = (\mathbf{p}^*, Z)$. L'ensemble $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}^*, \boldsymbol{\mathcal{Z}}\}$ définira par la suite une image augmentée, contenant intensité et carte de profondeur.

Soit une seconde image $\boldsymbol{\mathcal{I}}$ associée à une fonction d'intensité $\boldsymbol{\mathcal{I}}(\mathbf{p})$ et à un repère $\mathcal{F}$, ayant un déplacement 3D noté $\mathbf{T}(\tilde{\mathbf{x}}) \in \mathbb{SE}(3)$ exprimé dans le repère de l'image $\boldsymbol{\mathcal{I}}^*$ (cf. figure 2.2). Dans la suite de ce manuscrit l'image $\boldsymbol{\mathcal{I}}^*$ sera appelée image de référence et $\boldsymbol{\mathcal{I}}$ image courante.

Si la pose 3D $\mathbf{T}(\tilde{\mathbf{x}})$ entre le repère courant et le repère de référence est connue, il est possible de synthétiser une nouvelle vue, à partir des intensités $\boldsymbol{\mathcal{I}}(\mathbf{p})$ de l'image courante, à la position du repère de référence (Avidan & Shashua (1997)) en utilisant une fonction de *warping* :

$$\mathbf{p}^w = w(\mathbf{T}(\tilde{\mathbf{x}}); Z, \mathbf{p}^*). \tag{2.13}$$

La fonction $w(\cdot)$ transfère le pixel $\mathbf{p}^*$ de l'image de référence, associé au point 3D Euclidien $\mathbf{P} = (\mathbf{p}^*, Z)$, dans l'image courante par la transformation rigide $\mathbf{T}(\tilde{\mathbf{x}})$, suivie d'une projection dépendante du modèle de l'image courante (perspective *cf.* équation (2.10), omnidirectionnelle, sphérique *cf.* équation (3.26), cylindrique ...).

Cette fonction de *warping* $w(\mathbf{T}(\tilde{\mathbf{x}}); Z, \mathbf{p}^*) : \mathbb{SE}(3) \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$ est une action de groupe. Ce qui permet de définir les propriétés suivantes :

– L'application unité :

$$w(\mathbf{I}; Z, \mathbf{p}^*) = \mathbf{p}^*, \forall \mathbf{p}^* \in \mathbb{R}^2 \tag{2.14}$$

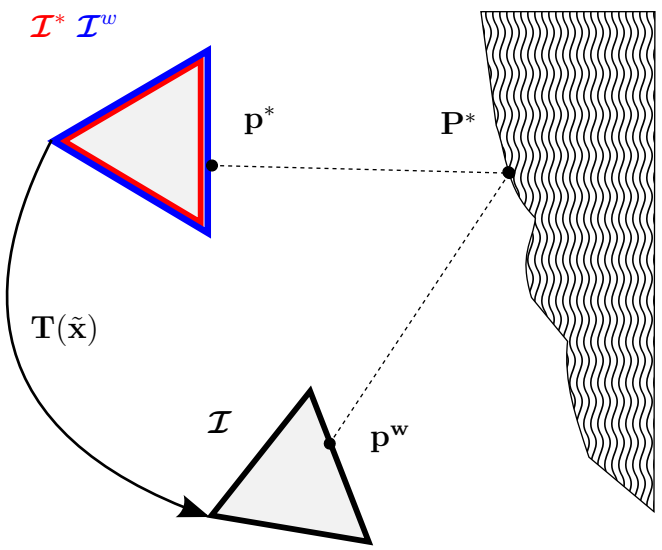


FIG. 2.2 – Position des caméras dans la scène. L'image $\mathcal{I}^*$ contient des intensités et l'information de profondeur. La caméra $\mathcal{I}$, observe la même scène pour un point de vue différent. Il est possible de synthétiser une nouvelle image $\mathcal{I}^w$ à partir de l'image $\mathcal{I}$, à la position de la caméra $\mathcal{I}^*$.

– La composition de deux actions correspond à l'action de la composition, c'est à dire, $\forall \mathbf{T_1}, \mathbf{T_2} \in \mathbb{SE}(3)$:

$$w(w(\mathbf{T_1}; Z, \mathbf{p}^*), \mathbf{T_2}) = w(\mathbf{T_1}\mathbf{T_2}; Z, \mathbf{p}^*), \forall \mathbf{p}^* \in \mathbb{R}^2 \tag{2.15}$$

2.4.2 Interpolations

Puisque en général, les points $\mathbf{p}^w$ ne correspondent pas directement à un pixel (c'est à dire $\mathbf{p}^w \notin \mathbb{N}^2$), la fonction d'intensité de l'image courante doit être interpolée aux coordonnées $\mathbf{p}^w$ pour obtenir l'intensité correspondante :

$$\mathcal{I}^w(\mathbf{p}^*) = \mathcal{I}(\mathbf{p}^w), \tag{2.16}$$

Dans la littérature, il existe plusieurs types d'interpolation :

– L'interpolation au plus proche voisin :

$$\mathcal{I}^w(\mathbf{p}^*) = \mathcal{I}(N(\mathbf{p}^w)), \tag{2.17}$$

où la fonction $N(.) : \mathbb{R}^2 \rightarrow \mathbb{N}^2$ calcule la valeur entière d'un réel. Cette méthode est la plus rapide en temps de calcul, mais des artéfacts (discontinuités) liés aux arrondis apparaissent sur les images transformées.

– L'interpolation bilinéaire (*cf.* figure 2.3), interpole les intensités sur un voisinage de 4 pixels. Soit la valeur entière $\mathbf{p}^N = N(\mathbf{p}^w)$, les coefficients d'interpolation bilinéaire sont alors définis par :

$$\boldsymbol{\alpha} = \mathbf{p}^w - \mathbf{p}^N = \begin{bmatrix} \alpha_u & \alpha_v \end{bmatrix} \in [0; 1], \tag{2.18}$$

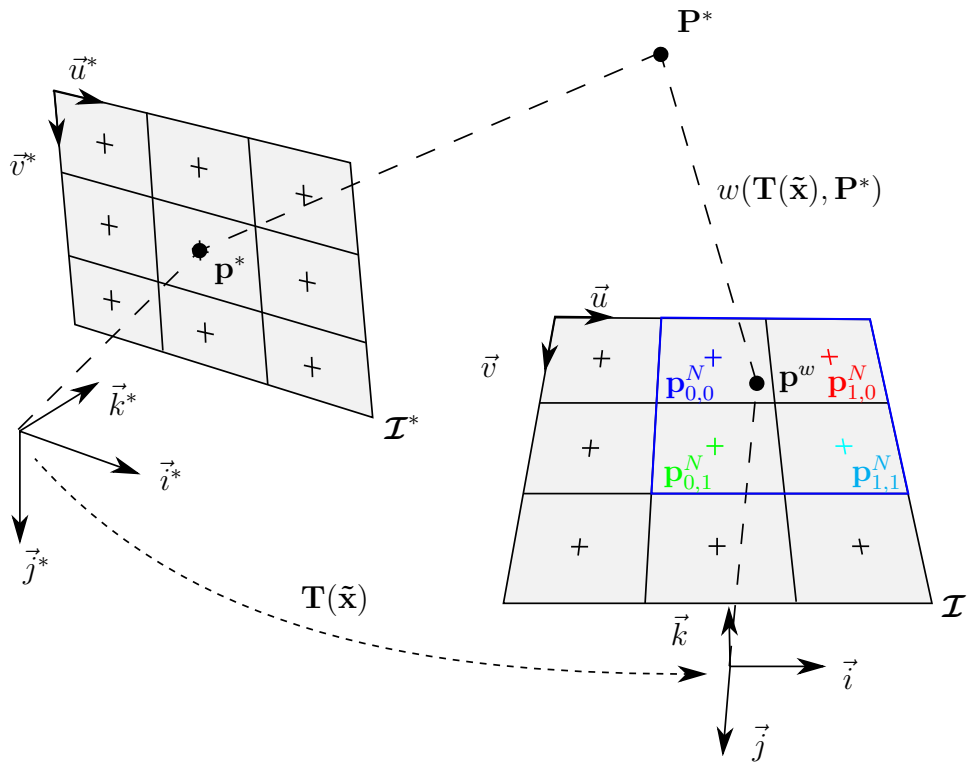


FIG. 2.3 – Interpolation bilinéaire. Le point 3D $\mathbf{P}^*$ associé au pixel $\mathbf{p}^*$ est projeté aux coordonnées $\mathbf{p}$ dans l'image $\mathcal{I}$.

Si l'on note $\mathbf{p}^N_{i,j} = \mathbf{p}^N + (i, j)$, la valeur interpolée aux coordonnées $\mathbf{p}^w$ s'écrit :

$$\mathcal{I}(\mathbf{p}^w) = \begin{bmatrix} 1-\alpha_v \\ \alpha_v \end{bmatrix}^T \begin{bmatrix} \mathcal{I}(\mathbf{p}^N_{0,0}) & \mathcal{I}(\mathbf{p}^N_{1,0}) \\ \mathcal{I}(\mathbf{p}^N_{0,1}) & \mathcal{I}(\mathbf{p}^N_{1,1}) \end{bmatrix} \begin{bmatrix} 1-\alpha_u \\ \alpha_u \end{bmatrix} \tag{2.19}$$

Cette méthode permet d'obtenir des images synthétisées plus lisses que l'interpolation au plus proche voisin.

– L'interpolation bicubique (*cf.* Keys (1981)), interpole les intensités sur un voisinage de 16 pixels :

$$\mathcal{I}(\mathbf{p}^w) = \begin{bmatrix} f(1+\alpha_v) \\ f(\alpha_v) \\ f(1-\alpha_v) \\ f(2-\alpha_v) \end{bmatrix}^T \begin{bmatrix} \mathcal{I}(\mathbf{p}^N_{-1,-1}) & \mathcal{I}(\mathbf{p}^N_{0,-1}) & \mathcal{I}(\mathbf{p}^N_{1,-1}) & \mathcal{I}(\mathbf{p}^N_{2,-1}) \\ \mathcal{I}(\mathbf{p}^N_{-1,0}) & \mathcal{I}(\mathbf{p}^N_{0,0}) & \mathcal{I}(\mathbf{p}^N_{1,0}) & \mathcal{I}(\mathbf{p}^N_{2,0}) \\ \mathcal{I}(\mathbf{p}^N_{-1,1}) & \mathcal{I}(\mathbf{p}^N_{0,1}) & \mathcal{I}(\mathbf{p}^N_{1,1}) & \mathcal{I}(\mathbf{p}^N_{2,1}) \\ \mathcal{I}(\mathbf{p}^N_{-1,2}) & \mathcal{I}(\mathbf{p}^N_{0,2}) & \mathcal{I}(\mathbf{p}^N_{1,2}) & \mathcal{I}(\mathbf{p}^N_{2,2}) \end{bmatrix} \begin{bmatrix} f(1+\alpha_u) \\ f(\alpha_u) \\ f(1-\alpha_u) \\ f(2-\alpha_u) \end{bmatrix} \tag{2.20}$$

où la fonction f peut être définie par une fonction sinus cardinal :

$$\mathrm{sinc}(\pi a) = \begin{cases} 0 & \text{si} \quad a = 0 \\ \frac{\sin(\pi a)}{\pi a} & \text{sinon} \end{cases} \tag{2.21}$$

Cette méthode permet d'obtenir des images lissées tout en préservant les contours, contrairement à l'interpolation bilinéaire qui a tendance à lisser les gradients des images.

– Il existe également des méthodes interpolant sur un voisinage de dimensions ≥ 64, à partir d'un filtre de Lanczos (*cf.* Duchon (1979)), cependant le temps de calcul

nécessaire pour ces approches devient rapidement trop important pour une utilisation temps-réel. D'autres méthodes, tels que le filtrage anisotrope, sont utilisées pour supprimer les effets d'*aliasing* liés au sur-échantillonnage, en préservant les contours, mais sont coûteuses en temps de calcul.

Dans ces travaux, l'interpolation bilinéaire a été utilisée, car elle offre un bon compromis entre temps de calcul et précision.

2.4.3 Hypothèse Lambertienne

Une hypothèse classique dans le domaine de la vision par ordinateur et notamment pour les méthodes directes d'alignement d'image, est que la scène observée est Lambertienne, ou conserve au moins des propriétés Lambertiennes localement autour d'un point de vue. En d'autres termes, chaque point 3D de la scène renvoie la même quantité de lumière quel que soit le point de vue. Dans ce cas, les changements d'intensité dans les images sont dus uniquement au déplacement des capteurs, il est donc possible d'écrire :

$$\boldsymbol{\mathcal{I}}^*(\mathbf{p}^*) = \boldsymbol{\mathcal{I}}\Big(w(\mathbf{T}(\tilde{\mathbf{x}}); Z, \mathbf{p}^*)\Big) \tag{2.22}$$

Cette équation signifie que les intensités de l'image courante transférées vers le repère de référence par la transformation rigide de changement de point de vue $\mathbf{T}(\tilde{\mathbf{x}})$ sont égales à celles de l'image originale.

2.5 Fonction d'erreur : SSD

Supposons maintenant que seulement une approximation $\widehat{\mathbf{T}}$ de $\mathbf{T}(\tilde{\mathbf{x}})$ est connue. Dans ce cas, le problème de recalage consiste à trouver la transformation incrémentale $\mathbf{T}(\mathbf{x})$:

$$\mathbf{T}(\tilde{\mathbf{x}}) = \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), \tag{2.23}$$

telle que les différences d'intensités entre les pixels de l'image courante recalée par la transformation $\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x})$ et celles de l'image de référence soient nulles :

$$\mathbf{e}(\mathbf{x}) = \boldsymbol{\mathcal{I}}\left(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); Z, \mathbf{p}^*)\right) - \boldsymbol{\mathcal{I}}^*(\mathbf{p}^*). \tag{2.24}$$

où $\mathbf{e}(\mathbf{x})$ est le vecteur de dimensions $mn \times 1$ contenant les erreurs associées à chaque pixel :

$$\mathbf{e}(\mathbf{x}) = \begin{bmatrix} e_1(\mathbf{x}) & e_2(\mathbf{x}) & \dots & e_n(\mathbf{x}) \end{bmatrix}^T \tag{2.25}$$

2.6 Minimisation efficace

2.6.1 Approximation du système d'équations

Puisqu'une approximation du déplacement $\mathbf{T}(\tilde{\mathbf{x}})$ est connue, on suppose que l'incrément $\mathbf{T}(\mathbf{x})$ est faible. Dans ce cas, il est possible de linéariser le vecteur $\mathbf{e}(\mathbf{x})$ en effectuant un développement en série de Taylor au voisinage de $\mathbf{x} = \mathbf{0}$:

$$\mathbf{e}(\mathbf{x}) = \mathbf{e}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0}, \mathbf{x})\mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3). \tag{2.26}$$

où $\mathbf{J}$ est la matrice Jacobienne du vecteur d'erreur $\mathbf{e}$, de dimensions $mn \times 6$ et représente le variation de $\mathbf{e}(\mathbf{x})$ en fonction de chaque composante de $\mathbf{x}$:

$$\mathbf{J}(\mathbf{x}) = \nabla_{\mathbf{x}}\mathbf{e}(\mathbf{x}) \tag{2.27}$$

et la matrice $\mathbf{M}(\mathbf{x_1}, \mathbf{x_2})$ de dimensions $mn \times 6$, est définie $\forall(\mathbf{x_1}, \mathbf{x_2}) \in \mathbb{R}^6 \times \mathbb{R}^6$ par :

$$\mathbf{M}(\mathbf{x_1}, \mathbf{x_2}) = \nabla_{\mathbf{x_1}}(\mathbf{J}(\mathbf{x_1})\mathbf{x_2}) = \left[\frac{\partial^2 e_1(\mathbf{x_1})}{\partial {\mathbf{x_1}}^2}\mathbf{x_2} \quad \frac{\partial^2 e_2(\mathbf{x_1})}{\partial {\mathbf{x_1}}^2}\mathbf{x_2} \quad \dots \quad \frac{\partial^2 e_n(\mathbf{x_1})}{\partial {\mathbf{x_1}}^2}\mathbf{x_2} \right]^T \tag{2.28}$$

où chaque matrice Hessienne $\frac{\partial^2 e_1(\mathbf{x_1})}{\partial {\mathbf{x_1}}^2}\mathbf{x_2}$ représente la dérivée seconde de $\mathbf{e}$ par rapport à $\mathbf{x}$.

2.6.2 Minimisation

Le système d'équations 2.26 peut être résolu avec une méthode des moindres carrés. Ce qui revient à minimiser la fonction de coût suivante :

$$\mathcal{O}(\mathbf{x}) = \frac{1}{2}\|\mathbf{e}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0}, \mathbf{x})\mathbf{x}\|^2, \tag{2.29}$$

Une condition nécessaire pour que le vecteur $\mathbf{x}$ soit un minimum de la fonction de coût est que la dérivée de $\mathcal{O}(\mathbf{x})$ soit nulle à la solution, c'est à dire en $\mathbf{x} = \tilde{\mathbf{x}}$:

$$\nabla_{\mathbf{x}}\mathcal{O}(\mathbf{x})|_{\mathbf{x}=\tilde{\mathbf{x}}} = \mathbf{0}, \tag{2.30}$$

Dans ce cas, la dérivée de la fonction de coût peut s'écrire :

$$\nabla_{\mathbf{x}}\mathcal{O}(\mathbf{x}) = (\mathbf{J}(\mathbf{0}) + \mathbf{M}(\mathbf{0}, \mathbf{x}))^T \left(\mathbf{e}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^2\right) \tag{2.31}$$

La méthode standard pour résoudre l'équation 2.30 est la méthode de Newton. Elle consiste à déterminer incrémentalement une solution $\mathbf{x}$ par :

$$\mathbf{x} = -\mathbf{Q}^{-1}\mathbf{J}(\mathbf{0})^T\mathbf{e}, \tag{2.32}$$

où la matrice $\mathbf{Q}$ s'écrit :

$$\mathbf{Q} = \mathbf{J}(\mathbf{0})^T\mathbf{J}(\mathbf{0}) + \sum_{i=0}^{n} \frac{\partial^2 e_i(\mathbf{x})}{\partial \mathbf{x}^2}\bigg|_{\mathbf{x}=\mathbf{0}} e_i \tag{2.33}$$

Cependant, la méthode de Newton nécessite le calcul des nm matrices Hessiennes, ce qui est couteux en temps de calcul. Il est cependant possible d'approcher la matrice $\mathbf{Q}$ avec une approximation au premier ordre par la méthode de Gauss-Newton :

$$\mathbf{Q} = \mathbf{J}(\mathbf{0})^T\mathbf{J}(\mathbf{0}) \tag{2.34}$$

Pour ce type d'approches, la méthode de Gauss-Newton est préférée, car elle permet d'une part d'assurer une matrice $\mathbf{Q}$ définie positive et d'autre part d'éviter le calcul assez coûteux des matrices Hessiennes.

Dans ces conditions, à chaque itération, une nouvelle erreur $\mathbf{e}$ et une nouvelle matrice Jacobienne $\mathbf{J}(\mathbf{0})$ sont calculées, afin d'obtenir la nouvelle valeur de $\mathbf{x}$ par :

$$\mathbf{x} = -\left(\mathbf{J}(\mathbf{0})^T\mathbf{J}(\mathbf{0})\right)^{-1}\mathbf{J}(\mathbf{0})^T\mathbf{e}(\mathbf{x}), \tag{2.35}$$

et mettre à jour la transformation rigide par :

$$\widehat{\mathbf{T}} \leftarrow \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}). \tag{2.36}$$

En général, la minimisation est arrêtée lorsque la norme de l'erreur : $\|\mathbf{e}\|^2 < \alpha$ ou lorsque l'incrément calculé devient trop faible : $\|\mathbf{x}\|^2 < \epsilon$, où α et ϵ sont des critères d'arrêt prédéfinis.

2.6.3 Inverse compositionnelle (IC)

Même si une méthode de type Gauss-Newton permet d'accélérer le temps de calcul et le domaine de convergence en évitant le calcul des matrices Hessiennes, il est néanmoins nécessaire d'évaluer la matrice Jacobienne $\mathbf{J}(\mathbf{0})$ à chaque itération ce qui peut être coûteux, notamment lorsque la dimension de l'image de référence est grande. La méthode d'inverse compositionnelle (Baker & Matthews (2001)), propose d'utiliser une matrice Jacobienne constante tout au long de la minimisation. Le gradient de l'erreur $\mathbf{J}(\mathbf{0})$, est approximé par la valeur du gradient à la solution, c'est à dire en $\mathbf{x} = \tilde{\mathbf{x}}$:

$$\mathbf{J}(\mathbf{0}) \approx \mathbf{J}(\tilde{\mathbf{x}}). \tag{2.37}$$

Puisqu'à la solution :

$$\boldsymbol{\mathcal{I}}\Big(w(\widehat{\mathbf{T}}\mathbf{T}(\tilde{\mathbf{x}}); Z, \mathbf{p}^*)\Big) = \boldsymbol{\mathcal{I}}^*(\mathbf{p}^*), \tag{2.38}$$

la matrice Jacobienne $\mathbf{J}(\tilde{\mathbf{x}})$ peut être pré-calculée sur l'image de référence. Cette matrice étant constante tout au long de la minimisation, l'algorithme devient alors très efficace en temps de calcul :

Algorithme 2.1 Inverse compositionnelle

Entrées: $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}^*, \boldsymbol{\mathcal{Z}}^*\}$, $\boldsymbol{\mathcal{I}}$, $\widehat{\mathbf{T}}$
  calculer la matrice Jacobienne $\mathbf{J}(\tilde{\mathbf{x}})$ de $\mathbf{S}$.
  itération $\leftarrow 1$
  répéter
    calculer l'image transformée $\boldsymbol{\mathcal{I}}^w$ (eq. 2.22).
    calculer le vecteur d'erreur $\mathbf{e}(\mathbf{x})$ (eq. 2.26).
    calculer l'incrément $\mathbf{x}$ (eq. 2.35).
    mettre à jour la pose : $\widehat{\mathbf{T}}$ (eq. 2.36).
    itération $\leftarrow$ itération $+1$
  jusqu'à $\|\mathbf{x}\| \leq \epsilon$ ou itération $>$ itération maximum
  Retourner $\widehat{\mathbf{T}}$.

2.6.4 Approximation du second ordre (*ESM*)

Dans Malis (2004); Benhimane & Malis (2004), la méthode de minimisation proposée permet d'obtenir des propriétés de convergence de second ordre, sans le calcul des matrices Hessiennes, tout en conservant une approximation de la matrice $\mathbf{Q}$ définie positive. Pour obtenir ce résultat, un développement en série de Taylor au premier ordre de la matrice $\mathbf{M}(\mathbf{0},\mathbf{x})$ de l'équation (2.26) est effectué, ce qui permet d'avoir une approximation du second ordre de la fonction d'erreur :

$$\mathbf{M}(\mathbf{0},\mathbf{x}) = \mathbf{J}(\mathbf{x}) - \mathbf{J}(\mathbf{0}) + \mathbf{O}(\|\mathbf{x}\|^2). \tag{2.39}$$

En remplaçant $\mathbf{M}(\mathbf{0},\mathbf{x})$ dans l'équation (2.26) on obtient :

$$\mathbf{e}(\mathbf{x}) = \mathbf{e}(\mathbf{0}) + \frac{1}{2}\left(\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x})\right)\mathbf{x} + \mathbf{O}(\|\mathbf{x}\|^3). \tag{2.40}$$

Lorsque $\mathbf{x} = \tilde{\mathbf{x}}$, l'équation devient :

$$\mathbf{e}(\tilde{\mathbf{x}}) \approx \mathbf{e}(\mathbf{0}) + \frac{1}{2}\left(\mathbf{J}(\mathbf{0}) + \mathbf{J}(\tilde{\mathbf{x}})\right)\tilde{\mathbf{x}} \tag{2.41}$$

La matrice $\mathbf{J}(\tilde{\mathbf{x}})$ correspond à la matrice Jacobienne obtenue avec l'algorithme d'inverse compositionnelle. La matrice $\mathbf{J}(\mathbf{0})$ est calculée à l'état courant de la minimisation, que l'on peut qualifier de *forward* compositionnelle. Ces deux matrices peuvent être divisées en 3 blocs (*cf.* détails en annexe V) :

$$\begin{aligned} \mathbf{J}(\tilde{\mathbf{x}}) &= \mathbf{J}_{\mathcal{I}^*}\mathbf{J}_w\mathbf{J}_\mathbf{T} \\ \mathbf{J}(\mathbf{0}) &= \mathbf{J}_{\mathcal{I}^w}\mathbf{J}_w\mathbf{J}_\mathbf{T} \end{aligned} \tag{2.42}$$

où $\mathbf{J}_{\mathcal{I}}$ est le gradient photométrique de l'image, de dimensions $mn \times 2mn$, $\mathbf{J}_w$ est la matrice Jacobienne de la fonction de *warping*, de dimensions $2mn \times 3$ et $\mathbf{J}_\mathbf{T}$ est la matrice Jacobienne de la paramétrisation de $\mathbf{x}$ de dimensions 3×6. Lors de la minimisation, seulement la matrice $\mathbf{J}_{\mathcal{I}^w} = \nabla\mathcal{I}^w$, correspondant au gradient photométrique de l'image courante transformée, a besoin d'être ré-évaluée. La mise à jour de la transformation $\mathbf{x}$ est alors obtenue d'une manière similaire à la méthode de Gauss-Newton par :

$$\mathbf{x} = -2\left(\mathbf{J}_{esm}^T\mathbf{J}_{esm}\right)^{-1}\mathbf{J}_{esm}^T\mathbf{e}(\mathbf{x}), \tag{2.43}$$

où la matrice $\mathbf{J}_{esm}$ est obtenue selon les équations (2.42) et (2.41) :

$$\mathbf{J}_{esm} = (\mathbf{J}_{\mathcal{I}^*} + \mathbf{J}_{\mathcal{I}^w})\mathbf{J}_w\mathbf{J}_\mathbf{T} \tag{2.44}$$

L'utilisation de l'algorithme *ESM*, permet une convergence quadratique vers la solution (*cf.* algorithme 2.2). Par rapport à une solution du premier ordre (*IC*) moins d'itérations sont nécessaires. Cependant, l'algorithme nécessite le calcul des gradients photométriques de l'image transformée à chaque itération de la minimisation, ce qui est couteux en temps de calcul. Dans les implémentations temps-réel des algorithmes, une itération de l'algorithme *ESM* coûte quasiment deux fois plus de temps qu'une itération de l'algorithme *IC*. La méthode *IC* peut donc effectuer deux fois plus d'itérations pour converger. Cependant, l'algorithme *ESM* présente d'autres avantages (*cf.* détails dans Benhimane (2006)), tel qu'une meilleure robustesse au sous échantillonnage et au bruit, et un domaine de convergence plus grand que les méthodes du premier ordre. Dans les expérimentations présentées dans les chapitres suivants, cet algorithme a été utilisé, cependant tous les algorithmes proposés restent identiques avec la méthode *IC*.

Algorithme 2.2 ESM

Entrées: $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}^*, \boldsymbol{\mathcal{Z}}^*\}$, $\boldsymbol{\mathcal{I}}$, $\widehat{\mathbf{T}}$.
  calculer les matrices Jacobienne $\mathbf{J}_{\boldsymbol{\mathcal{I}}^*}$, $\mathbf{J}_{\mathbf{w}}$ et $\mathbf{J}_{\mathbf{T}}$.
  itération $\leftarrow$ 1.
  répéter
    calculer l'image transformée $\boldsymbol{\mathcal{I}}^w$ (eq. (2.22)).
    calculer la matrice Jacobienne $\mathbf{J}_{\boldsymbol{\mathcal{I}}^{\mathbf{w}}}$ de $\boldsymbol{\mathcal{I}}^{\mathbf{w}}$.
    calculer la matrice Jacobienne $\mathbf{J}_{\mathbf{esm}}$ (eq. (2.44)).
    calculer le vecteur d'erreur $\mathbf{e}(\mathbf{x})$ (eq. (2.26)).
    calculer l'incrément $\mathbf{x}$ (eq. (2.43)).
    mettre à jour la pose : $\widehat{\mathbf{T}}$ (eq. (2.36)).
    itération $\leftarrow$ itération +1.
  jusqu'à $\|\mathbf{x}\| \leq \epsilon$ ou itération $>$ itération maximum.
  Retourner $\widehat{\mathbf{T}}$.

2.6.5 M-estimateurs

L'hypothèse de scènes entièrement Lambertiennes (*cf.* section 2.4.1) n'est en général pas vérifiée, en particulier pour des environnements 3D complexes (extérieurs ou intérieurs), où de nombreux objets ont des propriétés spéculaires. De plus, il est possible que des aberrations apparaissent dans les images transformées, les principales causes étant :

- Occultations partielles liées à la géométrie de l'environnement et du changement de point de vue entre les images (bâtiments, objets proches des caméras).
- Objets dynamiques (véhicules, piétons ...).
- Objets non rigides (végétation sous l'effet du vent...).
- Bruit des capteurs.
- Erreurs de modélisation : mauvaise information de profondeur sur les images de référence.

Bien que les techniques directes soient intrinsèquement robustes à ces erreurs, car l'information contenue dans les images est très redondante, des minimas locaux peuvent apparaitre lors de la minimisation : il est nécessaire de prendre ces aberrations en compte dans la fonction de coût.

De la même manière que dans Hager & Belhumeur (1998); Comport *et al.* (2010), il est possible d'utiliser une minimisation itérative robuste (*Iterative Re-weighted Least Squares* (*IRLS*)) pour gérer les aberrations à l'aide de M-estimateurs, détaillés dans Zhang (1995). La fonction de coût robuste peut alors s'écrire :

$$\mathcal{O}_\rho(\mathbf{x}) = \rho\left(\boldsymbol{\mathcal{I}}\left(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); Z, \mathbf{p}^*)\right) - \boldsymbol{\mathcal{I}}^*(\mathbf{p}^*)\right), \tag{2.45}$$

où la fonction $\rho(\mathbf{e})$ est une mesure pondérée de l'erreur $\mathbf{e}$. La solution robuste de $\mathbf{x}$ devient alors :

$$\mathbf{x} = -\left(\mathbf{J}^T\mathbf{D}\mathbf{J}\right)^{-1}\mathbf{J}^T\mathbf{D}\mathbf{e}. \tag{2.46}$$

où $\mathbf{D}$ est une matrice diagonale de dimensions $mn \times mn$:

$$\mathbf{D} = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_n \end{bmatrix} \tag{2.47}$$

contenant les poids $w_i \in [0;1]$, qui indiquent la confiance associée à chaque pixel telle que défini dans Huber (1981) :

$$w_i = \frac{\psi(\delta_i/\sigma)}{\delta_i/\sigma}, \quad \psi(u) = \begin{cases} u & , \text{ if } |u| \leq a \\ a\frac{u}{|u|} & , \text{ if } |u| > a, \end{cases} \tag{2.48}$$

où δ_i est le résidu centré de chaque valeur e_i par $\delta_i = e_i - Median(\mathbf{e})$ et ψ est la fonction d'influence. Le facteur de proportionnalité pour la fonction de Huber est $a = 1.345$, ce qui représente 95% d'efficacité dans le cas d'un bruit Gaussien.

Les valeurs δ_i sont normalisées par une mesure robuste de l'échelle de la distribution, la valeur absolue des écarts à la médiane (MAD : *Median absolute deviation*) :

$$\sigma = \frac{1}{\Phi^{-1}(0.75)} Median(|\delta_i - Median(\delta)|), \tag{2.49}$$

où $\Phi(\cdot)$ est la fonction de répartition et $1/\Phi^{-1}(0.75) = 1.48$ est la valeur de l'écart type pour une distribution normale.

Il existe dans la littérature d'autres fonctions robustes (telles que les fonctions de Cauchy, Tukey, Welsh, *etc*), cependant la fonction de Huber reste la plus efficace pour la majorité des cas (*cf.* Zhang (1995)).

La figure 2.4, montre le déroulement de la minimisation itérative robuste. A partir des profondeurs $\mathcal{Z}$ de l'image de référence augmentée $\mathcal{S}$, l'image $\mathcal{I}$ est recalé sur $\mathcal{I}^*$, ce qui permet de calculer une erreur d'intensité entre $\mathcal{I}^w$ et $\mathcal{I}^*$. L'estimateur robuste permet de rejeter les aberrations (ici réflexions spéculaires et occultations). Le mouvement $\widehat{\mathbf{T}}$ est alors mis à jour en calculant l'incrément $\mathbf{x}$. A chaque itération de l'optimisation, une nouvelle matrice $\mathbf{D}$ est recalculée en fonction de l'erreur $\mathbf{e}(\mathbf{x})$, jusqu'à convergence de l'algorithme.

2.7 Pyramide multi-résolution

Un inconvénient majeur des techniques directes et plus généralement des techniques de localisation itératives, est que l'approximation de la pose initiale $\widehat{\mathbf{T}}$ doit être assez proche de la solution $\mathbf{T}(\tilde{\mathbf{x}})$ pour converger (e.g à l'intérieur du bassin de convergence). Pour améliorer cette région de convergence, une approche multi-résolution est très souvent employée. Cette approche consiste à construire une pyramide de N images lissées et sous-échantillonnées successivement par un facteur 2 (*cf.* Burt & Adelson (1983)).

L'image du niveau $k+1$ de la pyramide est obtenue en sous-échantillonnant l'image $\mathcal{I}^k$, correspondant au niveau k, convoluée par un noyau Gaussien (lissage) :

$$\mathcal{I}^{k+1}(\mathbf{q}) = (\mathcal{I}^k(\mathbf{p}) \otimes \mathbf{G})(w(\mathbf{q})), \quad \forall \mathbf{q} = 2(\mathbf{p} - \mathbf{1}), \mathbf{p} \in \mathbb{N}^2, \tag{2.50}$$

où la fonction $w(\mathbf{q})$, sélectionne les coordonnées pixelliques paires de l'image convoluée $(\mathcal{I}^k(\mathbf{p}) \otimes \mathbf{G})$.

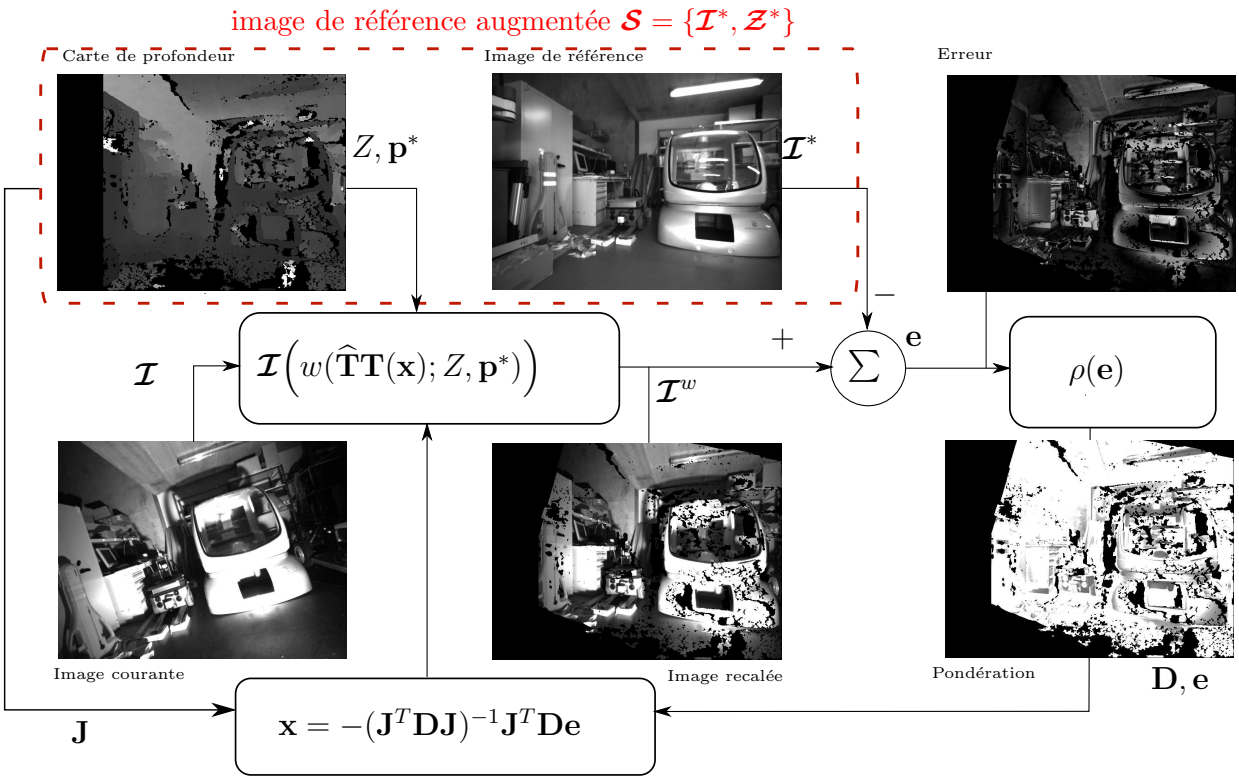


FIG. 2.4 – Processus de minimisation directe robuste : Les intensités de l'image $\boldsymbol{\mathcal{I}}$, sont itérativement recalées sur l'image $\boldsymbol{\mathcal{I}}^*$. L'image de pondération $\mathbf{D}$, calculée en fonction de l'erreur $\mathbf{e}$ permet mettre à jour le déplacement 3D $\mathbf{x}$ de manière robuste.

Le noyau Gaussien $\mathbf{G}$ est défini par :

$$\mathbf{G} = \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 34 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \tag{2.51}$$

Le niveau 0 de la pyramide, correspond à l'image originale $\boldsymbol{\mathcal{I}}^0$, de dimensions $m \times n$. Le reste de la pyramide est ensuite obtenu successivement à partir de l'équation de sous-échantillonnage (2.50) jusqu'au niveau $N-1$, où l'image $\boldsymbol{\mathcal{I}}^{N-1}$ est de dimensions $m/2^{N-1} \times n/2^{N-1}$.

Lors de l'alignement d'images, les pyramides des images photométriques $\boldsymbol{\mathcal{I}}$ et $\boldsymbol{\mathcal{I}}^*$ sont tout d'abord construites. Une pyramide de la carte de profondeur $\mathbf{Z}$ est également obtenue en utilisant un sous échantillonnage sans lissage Gaussien, de manière à préserver l'information géométrique. L'algorithme commence la minimisation à l'échelle $N-1$, correspondant aux images de plus petites dimensions, contenant très peu de détails. Après convergence, le résultat est ensuite utilisé pour initialiser le niveau suivant de la pyramide, et le recalage est de nouveau effectué. Cette opération est répétée jusqu'au niveau 0, correspondant à la plus grande résolution, donc à la meilleure précision.

Le fait de lisser les images de la pyramide, permet une convergence plus rapide vers le minimum de la fonction de coût, et aussi d'éviter certains minimas locaux supprimés par le lissage Gaussien. De plus, avec ce type d'approches, les grands déplacements sont minimisés à moindre coût sur la plus faible résolution, et la précision est obtenue sur la plus grande résolution en quelques itérations, ce qui permet d'accélérer le temps de calcul.

Algorithme 2.3 Suivi multi-résolution

Entrées: $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}^*, \boldsymbol{\mathcal{Z}}^*\}$, $\boldsymbol{\mathcal{I}}$, $\widehat{\mathbf{T}}$
  Construire la pyramide de $\boldsymbol{\mathcal{S}}$.
  Construire la pyramide de $\boldsymbol{\mathcal{I}}$.
  pour $k = N-1 \rightarrow 0$ **faire**
    Utiliser algorithme de suivi
    Si l'algorithme à convergé, mise à jour $\widehat{\mathbf{T}}$
  fin pour
  Retourner $\widehat{\mathbf{T}}$.

En pratique, pour une image de dimensions 800×600, 4 niveaux de pyramide sont utilisés (*cf.* figure 2.5). Pour une pyramide d'image perspectives, échantillonner d'un facteur α une image, modifie la matrice des paramètres intrinsèques par :

$$\mathbf{K}^{k+1} = \begin{bmatrix} 1/\alpha & 0 & 0 \\ 0 & 1/\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{K}^k. \tag{2.52}$$

2.8 Conclusion

La méthode de localisation visuelle 3D directe présentée dans ce chapitre, est basée sur une technique de synthèse de nouvelle vue, entre une image de référence augmentée par la

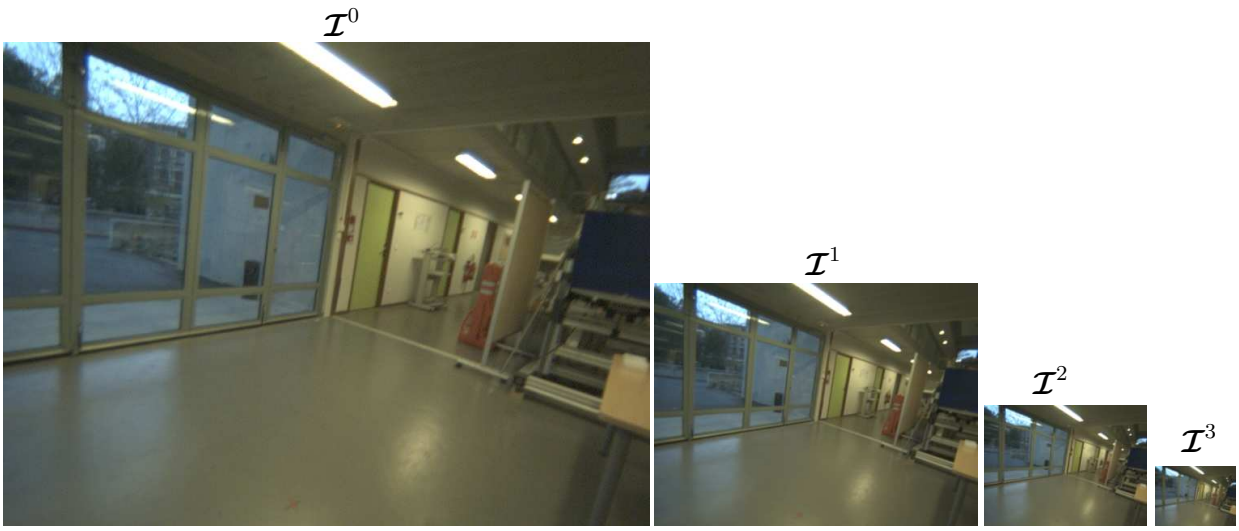


Fig. 2.5 – Pyramide multi-résolution.

profondeur et une image courante. La pose 3D entre les deux images peut être obtenue efficacement à l'aide d'une technique d'optimisation non linéaire, minimisant directement des erreurs d'intensités. Contrairement aux techniques basées points d'intérêts, tous les pixels des images sont utilisés pour la localisation, permettant une estimation précise et robuste de la pose. Les aberrations telles que les occultations et les changements d'illumination locaux sont directement pris en compte par une fonction robuste, alors qu'une approche multi-résolution permet d'accélérer la convergence de l'algorithme.

Dans la suite de ce manuscrit, les différentes notions définies dans ce chapitre, tel que la formation et la synthèse d'images, la minimisation robuste et les approches multi-résolution seront fréquemment employées.

Deuxième partie

Modélisation dense de l'environnement

Introduction

Comme introduit précédemment, l'acquisition et la construction automatique de modèles 3D denses, précis et utilisables pour la localisation n'est pas maitrisée à grande échelle. Pour obtenir une carte ré-utilisable pour une localisation précise, un modèle égo-centré est plus approprié. En effet conserver directement les images originales permet d'éviter les erreurs de reconstruction globales, liées aux approximations des modèles, et représente le plus fidèlement possible l'environnement local autour d'un point de vue.

Dans la littérature, toutes les approches de localisation quantitatives basées sur des mémoires images utilisent des techniques basées point d'intérêts (Royer *et al.* (2005); Courbon *et al.* (2008)). Afin de fournir une couverture visuelle maximale, certaines méthodes utilisent une représentation par images panoramiques (Jogan & Leonardis (2000); Cobzas *et al.* (2003)). Ce genre de représentation, déjà présent à très grande échelle dans l'application "grand public" Google Earth (*cf.* Vincent (2007)) permet une immersion visuelle photo-réaliste, mais dans ce cas n'est pas adaptée à la localisation précise, les sphères visuelles étant grossièrement positionnées dans l'espace.

Dans cette partie, une nouvelle approche alternative de cartographie est proposée. Pour cela, un modèle égo-centré sphérique est utilisé. Contrairement aux approches précédentes, cette représentation est *dense*, c'est à dire que tous les pixels des images peuvent être utilisés pour la localisation. Pour permettre l'utilisation de techniques de transfert d'images, nécessaires aux méthodes d'alignement d'images directes, l'augmentation de ces images avec l'information de profondeur associée à chaque pixel est indispensable. Cela amène à définir, ce qui sera appelé par la suite une sphère visuelle augmentée, telle que représentée sur la figure 2.6, et définie par :

$$\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}_S, \boldsymbol{\mathcal{Z}}_S, \boldsymbol{\mathcal{W}}_S, \boldsymbol{\mathcal{P}}_S\}, \tag{2.53}$$

- $\boldsymbol{\mathcal{I}}_S$ est la sphère photométrique contenant les intensités de chaque pixel.
- $\boldsymbol{\mathcal{P}}_S = \{\mathbf{q}_{S_1}, \ldots, \mathbf{q}_{S_n}\}$ est l'échantillonnage sur la sphère unité $\mathbf{q}_S \in S^2$.
- $\boldsymbol{\mathcal{Z}}_S$ est la carte de profondeur associée à chaque pixel de la sphère. Un point 3D est par conséquent défini sur la sphère par $\mathbf{P} = (\mathbf{q}_S, Z)$.
- $\boldsymbol{\mathcal{W}}_S$ est la carte de saillance permettant de sélectionner les meilleurs pixels de la sphère augmentée (*cf.* section 5.3).

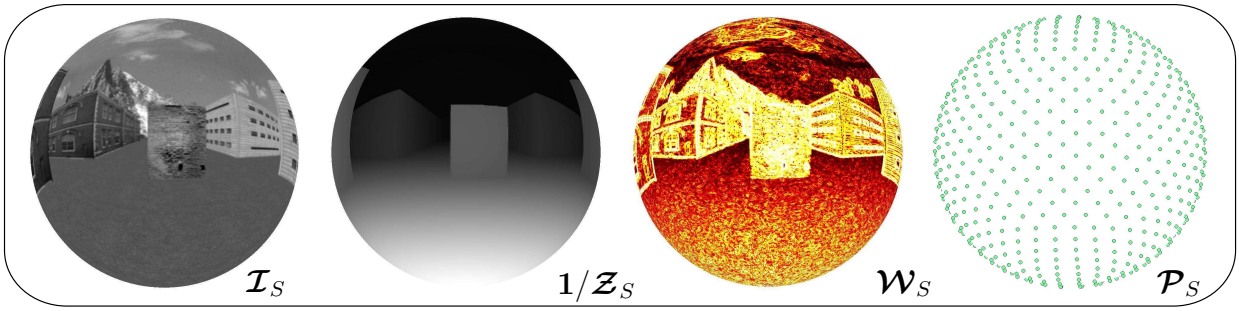


FIG. 2.6 – Représentation locale : sphère augmentée $\boldsymbol{\mathcal{S}}$ contenant des intensités $\boldsymbol{\mathcal{I}}_S$, une carte de profondeur $\boldsymbol{\mathcal{Z}}_S$, une carte de saillance $\boldsymbol{\mathcal{W}}_S$ et un échantillonnage spatial $\boldsymbol{\mathcal{P}}_S$.

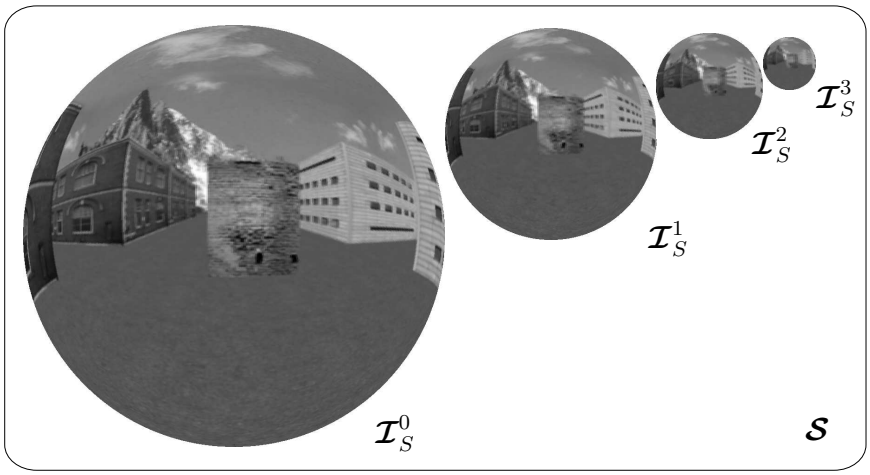


FIG. 2.7 – Sphères multi-résolution.

Pour permettre l'utilisation de techniques d'alignement d'images multi-résolution, la sphère visuelle augmentée $\boldsymbol{\mathcal{S}}$ est également décomposée en pyramide multi-résolution : $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{S}}^0, \boldsymbol{\mathcal{S}}^1, \ldots, \boldsymbol{\mathcal{S}}^N\}$, construite à partir de lissages et de sous-échantillonnages successifs de la résolution de base tel que défini dans la section 2.7 et montré sur la figure 2.7.

Cette sphère est le modèle local de la carte, elle permet de localiser avec une méthode d'alignement directe, une caméra, naviguant localement dans son voisinage. Pour permettre une localisation globale, il est nécessaire de définir le modèle global, c'est à dire le graphe $\mathcal{G}$, représenté sur la figure 2.8, et défini tel que

$$\mathcal{G} = \{\boldsymbol{\mathcal{S}}_1, \ldots, \boldsymbol{\mathcal{S}}_n; \mathbf{T}_1, \ldots, \mathbf{T}_m\} \tag{2.54}$$

où $\boldsymbol{\mathcal{S}}_1, \ldots, \boldsymbol{\mathcal{S}}_n$ est l'ensemble des n sphères augmentées connectées rigidement entre elles par les m poses : $\mathbf{T}_1, \ldots, \mathbf{T}_m$ (*cf.* figure 2.8).

Cette nouvelle représentation comporte de nombreux avantages :

- Une représentation égo-centrée permet de conserver des mesures directement extraites des capteurs. Cela assure localement une précision maximale (par rapport aux données initiales). De plus toute l'information nécessaire pour la localisation 3D est présente et compactée dans une seule sphère augmentée, ce qui évite de cartographier les zones inutiles à la navigation (*i.e.* zones hors voies de circulation).
- L'information dense de profondeur permet d'utiliser des techniques directes d'estimation de pose à 6 degrés de liberté, basées sur le transfert d'images (*cf.* section 2.4.1).
- Contrairement aux modèles *CAO* texturés, la consistance photométrique d'une image améliore les performances des techniques directes de recalage d'image (vitesse et bassin de convergence), mais aussi la robustesse des techniques basées point d'intérêts, sensibles aux changements de points de vue.
- La généricité d'une représentation sphérique permet de combiner différents capteurs pour la localisation, tel que les caméras perspectives, les caméras stéréoscopiques, les caméras omnidirectionnelles ou les capteurs laser.
- Un capteur à large angle de vue augmente l'observabilité des mouvements 3D (*cf.* Baker *et al.* (2001)).

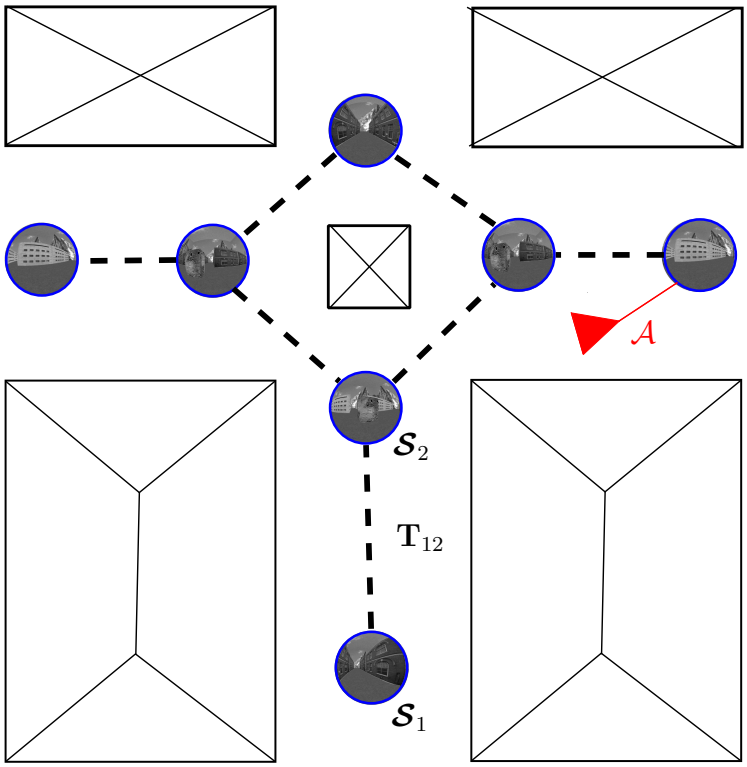


FIG. 2.8 – Représentation égo-centrée : graphe de sphères augmentées $\mathcal{G}$ permettant la localisation d'un agent $\mathcal{A}$ naviguant localement à l'intérieur du graphe.

– Une seule sphère de vision permet de cartographier par exemple une route à double-sens de circulation, permettant d'avoir un modèle plus compact.

Cependant, la construction d'un tel modèle a certaines limitations. Tout d'abord, il n'existe aucun capteur permettant l'acquisition de sphères visuelles augmentées par la profondeur. Cela nécessite de développer des capteurs et des algorithmes spécialement conçus pour cette tâche. Le premier chapitre de cette partie présente les différentes méthodes de construction d'images panoramiques et un nouveau capteur permettant la construction de sphères visuelles augmentées est proposé. Avec ce système, il est possible de reconstruire des images panoramiques augmentées par la profondeur, sans utiliser de capteur actif.

Dans le deuxième chapitre, une méthode de positionnement et de sélection automatique des sphères du graphe est présentée. Les résultats expérimentaux montreront qu'avec un tel système, il est possible de cartographier de manière dense et compacte, de larges environnements.

Comme il a été présenté en première partie, les méthodes directes d'alignement d'images utilisent tous les pixels présents dans les images dans une boucle de minimisation itérative. Le graphe d'images sphériques étant destiné à être utilisé en temps réel pour la localisation, la quantité de pixels présents dans les sphères peut être difficile à traiter en temps réel. Le troisième chapitre propose une nouvelle méthode de sélection de pixels saillants, pré-calculée lors de la construction de la base de données et permettant d'utiliser un nombre réduit de pixels lors de la localisation en ligne, sans dégrader l'observabilité des mouvements 3D.

Chapitre 3

Construction d'une sphère augmentée

3.1 Systèmes existants

3.1.1 Sphère photométrique panoramique

Alors que les travaux de Krishnan & Nayar (2009) présentent un vrai capteur sphérique, l'acquisition d'images panoramiques de bonne qualité est encore un problème non résolu. En effet, seule la construction de capteurs planaires (CCD ou CMOS) est maitrisée, et donc de caméras à champ de vue limité. Bien que certains objectifs permettent l'acquisition d'images très grand angle (*e.g.* objectif fisheye, *cf.* figure 3.1), la vision panoramique à 360° est d'une manière générale simulée. Deux techniques sont majoritairement utilisées : les caméras omnidirectionnelles catadioptriques et les systèmes multi-caméras.

3.1.1.1 Caméra omnidirectionnelle catadioptrique

Les caméras omnidirectionnelles Nayar (1997) permettent d'acquérir des images avec un champ de vision horizontal de 360° et avec un centre de projection unique. Ce capteur est en général composé d'une caméra perspective classique ou orthographique, à laquelle vient se greffer un miroir convexe de forme le plus souvent hyperbolique ou parabolique (voir figure 3.2), placé dans l'axe optique de la caméra.

Ce type de caméra peut être modélisé par le modèle de projection unifié proposé par Mei & Rives (2007), qui est une extension de Geyer & Daniilidis (2000). Ce modèle effectue deux projections successives : une projection sphérique suivie d'une projection perspective (*cf.* figure 3.3). Un point $\mathbf{P} \in \mathbb{R}^3$ de l'espace Euclidien est projeté sur la sphère unitaire par une projection sphérique :

$$\mathbf{q}_S = \frac{\mathbf{P}}{\|\mathbf{P}\|}. \tag{3.1}$$

Le point $\mathbf{q}_S$ est exprimé dans le repère $\mathcal{F}_M$ par :

$$\mathbf{q}_M = \mathbf{q}_S + \mathbf{e}_3\xi \tag{3.2}$$

où le paramètre $\xi \in [0; 1]$ dépend de la géométrie du miroir. Le point $\mathbf{q}_M$ est alors projeté dans le plan normalisé de la caméra par :

$$\overline{\mathbf{m}} = \frac{\mathbf{q}_M}{\mathbf{e}_3^T\mathbf{q}_M}, \tag{3.3}$$

FIG. 3.1 – Objectif grand angle et image fisheye.

FIG. 3.2 – Caméra omnidirectionnelle catadioptrique.

et dans le plan image normalisé par :

$$\overline{\mathbf{p}} = \mathbf{K}\overline{\mathbf{m}} = \begin{bmatrix} \gamma_1 & \gamma_1 s & u_0 \\ 0 & \gamma_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \overline{\mathbf{m}}, \tag{3.4}$$

où $\mathbf{K}$ est la matrice des paramètres intrinsèques de la caméra, et (γ_1, γ_2) sont les distances focales généralisées et dépendent de la forme du miroir. Les valeurs théoriques de γ et ξ sont détaillées dans Mei & Rives (2007), et peuvent être déterminées lors d'une phase d'étalonnage.

L'image omnidirectionnelle $\boldsymbol{\mathcal{I}}_o$ peut ainsi être re-projetée sur une image sphérique $\boldsymbol{\mathcal{I}}_S$ au moyen d'un *warping* stéréographique inverse :

$$\boldsymbol{\mathcal{I}}_S(\mathbf{q}_S) = \boldsymbol{\mathcal{I}}_o(w(\mathbf{K}, \xi, \mathbf{q}_S)) \tag{3.5}$$

L'inconvénient majeur de ce genre de caméras est d'une part, une faible résolution (360° sont projetés sur un seul capteur perspectif), et d'autre part une résolution spatiale non

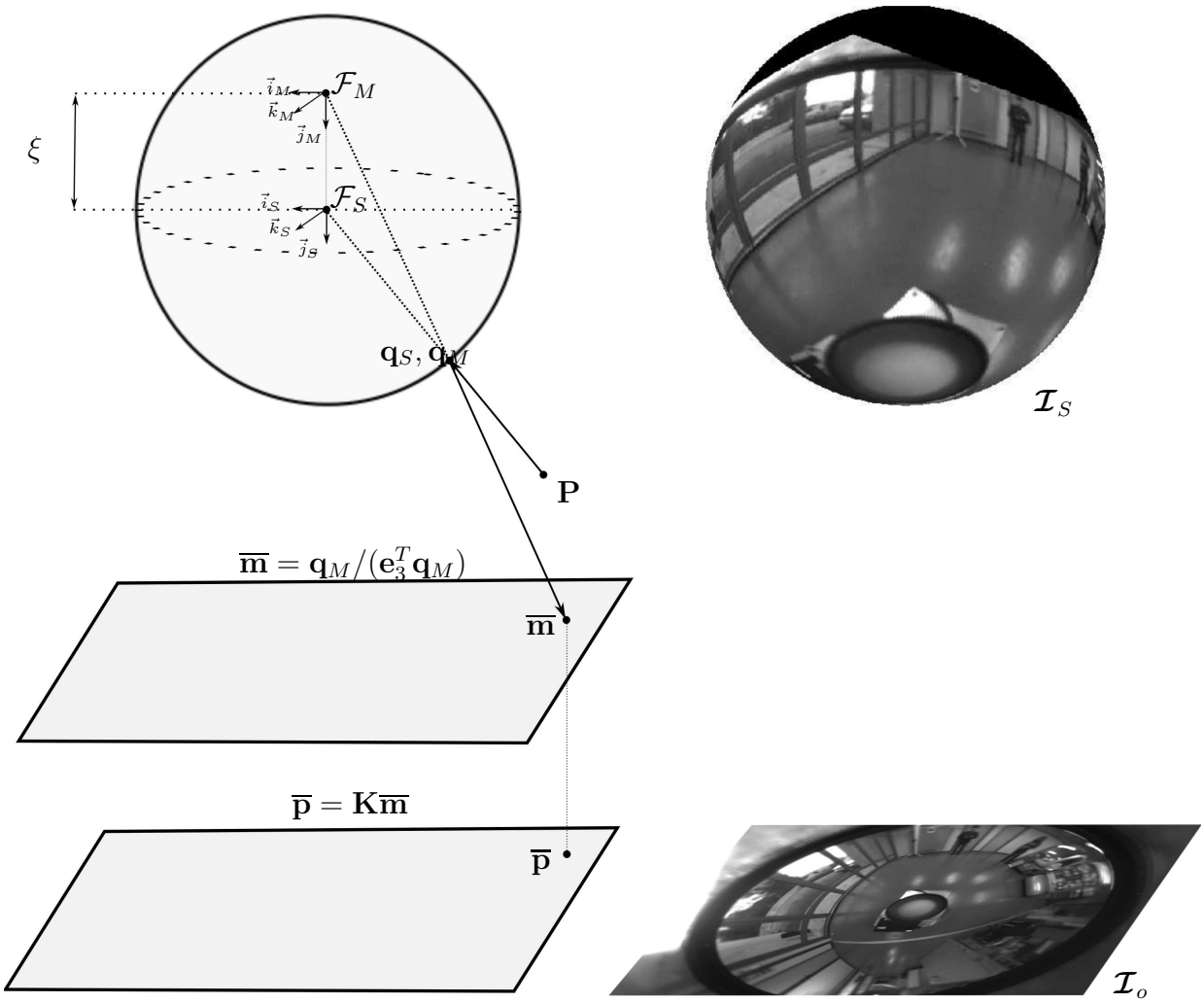


FIG. 3.3 – Modèle de projection unifié.

uniforme : la qualité de l'image diminue en direction des bords du capteur (*cf.* figure 3.3). De plus en fonction de la forme du miroir, la largeur en champ vertical est limitée (demi-sphère,<90°), ce qui n'est pas idéal pour cartographier des environnements urbains : la partie saillante et stable de l'information se trouvant souvent en hauteur (façade des bâtiments).

3.1.1.2 Systèmes multi-caméra

Il est également possible de construire une image panoramique, assemblée à partir de plusieurs images, capturées simultanément par plusieurs caméras reliées rigidement Baker *et al.* (2001) ou issues d'une séquence d'image Lovegrove & Davison (2010). Les images, dont la position doit être parfaitement connue sont alors recalées, projetées et fusionnées sur une sphère virtuelle tangente aux capteurs par une technique de *mosaicing* Szeliski (2006). Les N images $\boldsymbol{\mathcal{I}}_i$ peuvent être transformées et fusionnées sur une sphère par une fonction de *warping* des intensités des images perspectives vers l'image sphérique $\boldsymbol{\mathcal{I}}_S$:

$$\boldsymbol{\mathcal{I}}_S(\mathbf{q}_S) = \boldsymbol{\alpha}_1\boldsymbol{\mathcal{I}}_1\Big(w(\mathbf{K}_1, \mathbf{R}_1, \mathbf{q}_S)\Big) + \ldots + \boldsymbol{\alpha}_N\boldsymbol{\mathcal{I}}_N\Big(w(\mathbf{K}_N, \mathbf{R}_N, \mathbf{q}_S)\Big), \tag{3.6}$$

où les cœfficients $\boldsymbol{\alpha}$ sont les cœfficients de fusion des intensités, les matrices $\mathbf{K}_i$ les paramètres intrinsèques des caméras et les matrices $\mathbf{R}_i$ représente la rotation des images par rapport à la sphère. Puisque l'information de profondeur n'est pas connue, les translations $\mathbf{t}_i$, entre les images et la sphère sont obligatoirement négligées. La fonction $\overline{\mathbf{p}} = w(\mathbf{K}, \mathbf{R}, \mathbf{q_S})$ transfère le point de la sphère unitaire $\mathbf{q}_S \in \mathcal{S}^2$ dans l'image par une projection perspective (*cf.* figure 3.4(c)) :

$$\overline{\mathbf{p}} = \frac{\mathbf{K}\mathbf{R}\mathbf{q}_S}{\mathbf{e}_3^T\mathbf{K}\mathbf{R}\mathbf{q}_S}. \tag{3.7}$$

Grâce à l'utilisation de plusieurs images issues de capteurs perspectifs, ce genre de technique permet de construire des images sphériques de très grande résolution (>10 millions de pixels). Néanmoins, le fait de négliger les translations, revient à assumer un centre de projection commun à toutes les caméras afin d'aligner les images en rotation uniquement (la rotation étant indépendante de la géométrie de la scène). Dans ce cas les centres optiques doivent être le plus proche possible les uns des autres, ce qui peut être problématique en terme de conception mécanique car le centre optique d'une caméra est un point virtuel.

Dans certains cas, notamment lorsque des objets de la scène sont proches des capteurs, la translation entre les centres optiques n'est pas négligeable, l'hypothèse du centre de projection unique n'est pas valable : des artéfacts liés aux effets de parallaxe sont visibles dans les images panoramiques reconstruites.

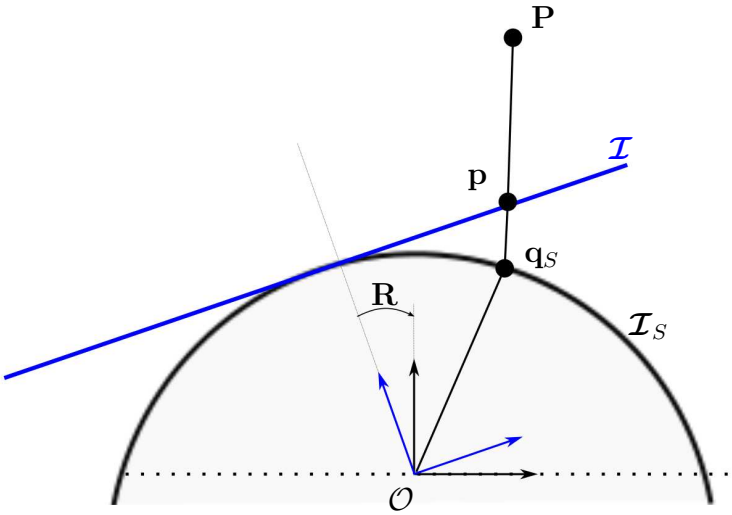
Pour minimiser cet effet de parallaxe, Li (2006a) a proposé une caméra sphérique composée de deux objectifs fisheye placés dos à dos. L'image finale est formée sur un seul capteur à l'aide d'un miroir. Ce système permet de minimiser la translation entre les deux caméras virtuelles et ainsi obtenir un centre de projection quasiment unique. Cependant, le fait de n'utiliser qu'un seul capteur ne permet pas d'obtenir des sphères de grande résolution.

3.1.2 Sphère de profondeur

Les systèmes d'acquisition d'images sphériques actuels ne permettent pas d'extraire la profondeur à partir d'une seule image. Il est cependant possible d'utiliser deux capteurs "sphériques" pour appliquer des techniques de mise en correspondance dense stéréo (*cf.* Hirschmuller (2008)). Une autre catégorie de capteur dit hybride consiste à associer un capteur actif de type télémètre laser à une caméra sphérique.

(a) Caméras sphériques. *PointGrey,Pfeil* et al. *(2011) et Nikon.*

(b) Image panoramique. *LadyBug PointGrey.*



(c) *Mosaicing*

FIG. 3.4 – (a) : Exemples de systèmes d'acquisition d'images panoramique multi-caméra .(b) Image panoramique reconstruite. (c) : Transformation d'une image perspective sur la sphère.

3.1.2.1 Systèmes passifs

Dans Li (2006b), des techniques de mise en correspondance dense stéréo sont appliquées à deux images sphériques provenant de caméras fisheye. Caron *et al.* (2011) extrait l'information de profondeur à partir d'une caméra omnidirectionnelle composée de 4 miroirs. Encore une fois, les techniques basées sur les caméras omnidirectionnelles ont une mauvaise résolution spatiale et ne sont pas adaptées aux environnements urbains.

Récemment Kim & Hilton (2009) ont utilisé deux caméras mono-dimensionnelles pivotantes en configuration verticale pour reconstruire deux images sphériques. La profondeur est ensuite obtenue par mise en correspondance dense. Cependant ce genre de techniques, basées sur des capteurs en mouvements, est inadaptée aux scènes dynamiques et donc difficilement embarquable sur des véhicules mobiles.

3.1.2.2 Systèmes actifs

Dans Gallegos *et al.* (2010) une image sphérique est construite avec une caméra omnidirectionnelle classique. L'information de profondeur est obtenue à partir d'un télémètre laser placé au dessus de la caméra. La propagation de l'information de profondeur dans l'image nécessite l'hypothèse d'un sol planaire et d'un environnement structuré contenant des murs verticaux. Dans Cobzas *et al.* (2003) une idée similaire est utilisée mais une caméra perspective est montée avec un télémètre laser sur une tourelle pan/tilt. Une rotation de 360 °du système permet d'obtenir une image cylindrique augmentée par l'information dense de profondeur. Encore une fois la vitesse de rotation du système ne permet pas de l'embarquer sur un véhicule mobile.

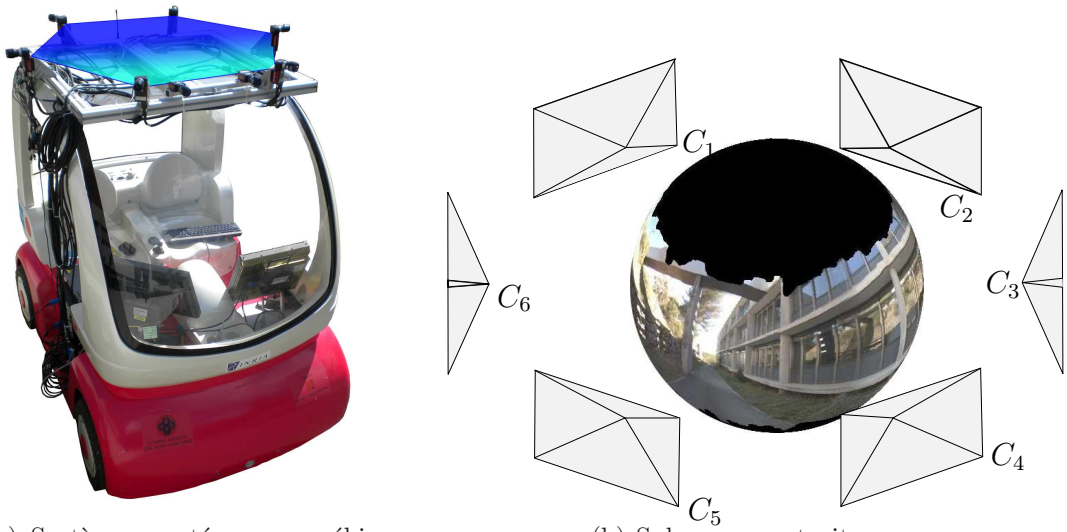
Actuellement très populaires dans la communauté de vision pour la robotique, les caméras RGB+D, basées sur la projection de lumière structurée (souvent infra-rouge), permettent d'obtenir des images augmentées par la profondeur en temps réel et sont alors utilisées dans des systèmes de *SLAM* tels que Audras *et al.* (2011); Newcombe *et al.* (2011a); Henry *et al.* (2010). Dans Spinello & Arras (2011) 3 caméras RGB+D sont utilisées pour obtenir une image panoramique. Toutefois ces systèmes sont prévus pour des environnements intérieurs et ne sont pas utilisables à l'extérieur car très sensibles à la lumière du soleil.

3.2 Système d'acquisition de sphères augmentées

3.2.1 Système de caméras à *multi-baselines*

Le système d'acquisition développé dans cette thèse utilise six caméras grand angle placées sur un cercle dont les centres optiques sont éloignés volontairement les uns des autres. Contrairement aux systèmes multi-caméras classiques, cette configuration permet de générer de la disparité entre chaque image et ainsi utiliser des techniques de mise en correspondance dense stéréo pour extraire la profondeur, directement sur les images du système. Cette information est d'une part indispensable pour une localisation à 6 degrés de liberté, et d'autre part pour créer une sphère à centre de projection unique. En effet, l'information de profondeur permet de re-projeter correctement la photométrie issue des images, ce qui évite les artéfacts liés aux effets de parallaxe dont souffrent les systèmes panoramiques multi-caméras classiques.

La figure 3.5 montre le système monté sur un véhicule. Dans cette configuration, les 360° du champ de vision horizontal sont visibles par le système. Grâce à la large *baseline*



(a) Système monté sur un véhicule.

(b) Sphere reconstruite.

FIG. 3.5 – Système d'acquisition de sphères augmentées. La disposition hexagonale des caméras permet l'utilisation de techniques de mise en correspondance dense.

séparant chaque caméra, et aux objectifs grands angles utilisés, une disparité peut être extraite dans chaque image.

3.2.2 Étalonnage

Avant de pouvoir effectuer la mise en correspondance dense et ainsi reconstruire des sphères visuelles augmentées, il est important de calibrer le système. C'est à dire extraire les paramètres extrinsèques (position relative des caméras) et les paramètres intrinsèques des caméras (focale, centre optique, polynôme de distorsions). Le système multi-caméras proposé ici peut être représenté comme 6 paires caméras stéréo, où chaque paire stéréo est reliée rigidement à la paire suivante.

La particularité de ce dispositif est que l'angle formé entre les axes optiques de chaque caméra est divergent (60°). Dans ces conditions une mire classique d'étalonnage (échiquier) ne peut être observée que par deux caméras simultanément, ce qui ne permet pas d'utiliser des techniques d'étalonnage multi-caméras telles que Svoboda *et al.* (2005); Zaharescu *et al.* (2006) qui assument l'objet d'étalonnage visible par toutes les caméras. D'autres techniques Li (2006a) utilisent une seule mire rigide englobant le système afin d'étalonner toutes les caméras simultanément. Cependant ce genre d'approche est difficilement applicable au système présenté ici, en particulier à cause de l'échelle : une mire rigide de plusieurs mètres est nécessaire.

Afin d'assurer une mise en œuvre simple, une méthode d'étalonnage utilisant une mire simple (objet plan) a été développée. La technique la plus basique consiste à estimer successivement les paramètres extrinsèques des caméras avec un étalonnage stéréo classique Bouguet (2005). Dans ce cas, les erreurs sont cumulées, résultant en un étalonnage inconsistant : la dernière paire stéréo contiendra toute la dérive intégrée sur chaque paire stéréo.

Cependant la configuration circulaire du système présente une fermeture de boucle (cf. Fig. 3.6). Dans ces conditions il est possible de formuler le problème en une optimisation globale des paramètres extrinsèques du système ainsi que des poses des mires d'étalonnage.

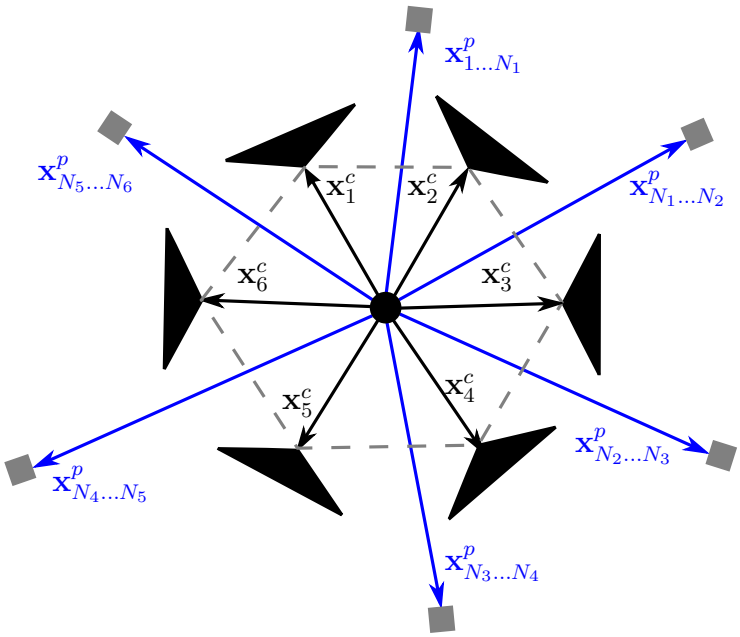


FIG. 3.6 – Étalonnage du système d'acquisition sphérique.

Cela permet de corriger la dérive et de répartir les erreurs de re-projection sur toutes les caméras.

Le vecteur des inconnues du système est défini tel que :

$$\mathbf{x}^{\Sigma} = (\mathbf{x}^c_1, ..., \mathbf{x}^c_M, \mathbf{x}^p_1, ..., \mathbf{x}^p_N)^{\top} \tag{3.8}$$

où $\mathbf{x}^c_M$ représente les poses des caméras ($M = 6$) et $\mathbf{x}^p_N$ représente les N poses des mires.

Le critère d'optimisation global est défini (avec abus de notation) par l'erreur entre le vecteur des points de la mire projetée $w(\boldsymbol{\mathcal{P}}_p)$ et le vecteur des points détectés dans les images $\boldsymbol{\mathcal{P}}_m$:

$$\mathbf{e}(i,j) = \boldsymbol{\mathcal{P}}_m - w\left(\mathbf{T}(\mathbf{x}^c_i)\mathbf{T}(\mathbf{x}^p_j), \boldsymbol{\xi}_i; \boldsymbol{\mathcal{P}}_p, \boldsymbol{\mathcal{Z}}_p\right), \tag{3.9}$$

où i et j sont respectivement l'indice de la caméra et l'indice de la mire (voir Fig. 3.6). La matrice $\mathbf{K}(\boldsymbol{\xi}_i) \in \mathbb{R}^{3\times 3}$ contient les paramètres intrinsèques de la caméra i. Dans ce cas, la fonction $w(.)$ est une projection perspective qui transfère les points de la mire j sur la caméra i :

$$\overline{\mathbf{p}} = \frac{\boldsymbol{K}\left[\begin{array}{cc}\mathbf{R} & \mathbf{t}\end{array}\right]\mathbf{P}}{\mathbf{e}_3^T\boldsymbol{K}\left[\begin{array}{cc}\mathbf{R} & \mathbf{t}\end{array}\right]\mathbf{P}}, \tag{3.10}$$

où $\mathbf{P}$ est un point 3D Euclidien appartenant à la mire de calibration.

A partir de cette fonction d'erreur, il est possible de trouver un jeu de paramètres optimal $\widehat{\mathbf{x}}^{\Sigma}$ en minimisant l'erreur de re-projection pour chaque caméra et chaque mire :

$$\widehat{\mathbf{x}}^{\Sigma} = \arg\min_{\mathbf{x}^{\boldsymbol{\Sigma}}} \sum_{i=1}^{6}\left(\sum_{j=1}^{N}\|\mathbf{e}(i,j)\|^2.\eta(i,j)\right) \tag{3.11}$$

$$\eta(i,j) = \begin{cases} 1 & \text{si la mire j est vue par la caméra i} \\ 0 & \text{sinon} \end{cases}$$

Minimiser itérativement la fonction de coût (3.11) permet d'estimer la pose de chaque caméra $\mathbf{x}_i^c$ en respectant la contrainte de fermeture de boucle. En pratique, afin d'éviter certains minimas locaux, la minimisation est initialisée avec les paramètres extrinsèques obtenus successivement par étalonnage stéréo Bouguet (2005). Les paramètres intrinsèques $\boldsymbol{\xi}_k$ quand à eux peuvent être obtenus indépendamment et précisément pour chaque caméra, ils ne sont donc pas re-estimés dans la minimisation.

3.2.2.1 Résultats

Un étalonnage du système a été effectuée. En raison des grands angles des caméras et de la *baseline* du système, une mire de dimensions A0 (841 x 1189 mm) a été utilisée afin d'assurer une détection précise de l'échiquier d'étalonnage.

La figure 3.7 montre l'évolution de l'erreur de re-projection de l'équation (3.11) pour 25 itérations : l'algorithme converge rapidement vers le minimum de la fonction de coût. Les valeurs de l'erreur à la fermeture de boucle, avant et après étalonnage global sont montrées dans le tableau 3.1. L'optimisation globale a permis de réduire l'erreur totale.

Étape	Translations (X,Y,Z) (mm)	Rotations (X,Y,Z) (degrés)
Initialisation	$(-14.9750, 6.1387, -3.2574)$	$(-0.4669, 0.0300, 0.1333)$
Optimisation globale	$(-1.5619, 2.3673, -1.9614)$	$(0.1137, 0.0053, -0.4547) \times 10^{-3}$

Tab. 3.1 – Erreur sur les paramètres extrinsèques à la fermeture de boucle avant et après étalonnage global.

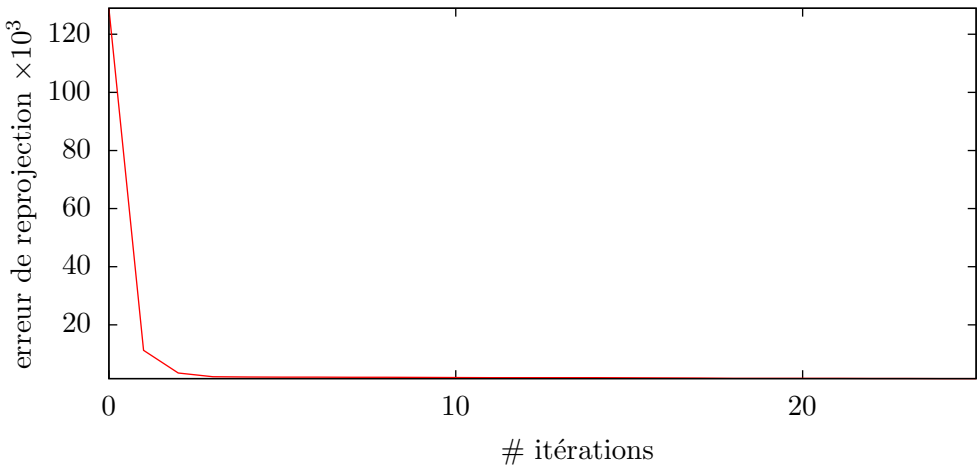


Fig. 3.7 – Étalonnage du système d'acquisition : évolution de l'erreur de re-projection.

3.2.3 Extraction de la profondeur

3.2.3.1 Rectification des images

Pour une paire stéréo calibrée, c'est à dire lorsque les paramètres intrinsèques et extrinsèques sont connus, il est possible de rectifier les images afin que la mise en correspondance dense soit ramenée à une recherche mono-dimensionnelle le long des lignes épipolaires.

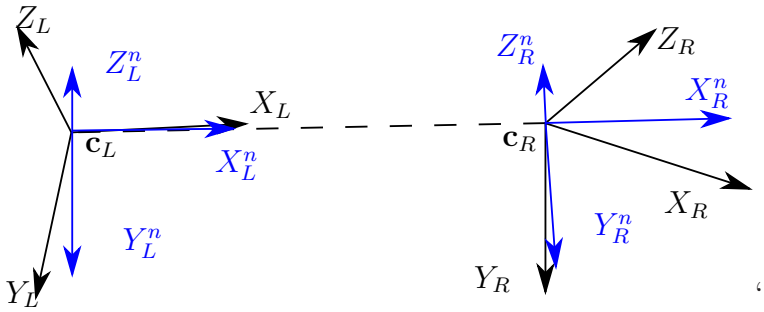


FIG. 3.8 – Repères de rectification stéréo.

La technique proposée dans Fusiello *et al.* (2000) permet de générer un nouveau couple d'images stéréo parfaitement fronto-parallèle, en appliquant une rotation autour des centres optiques de chaque caméra. La transformation effectuée assure des épipoles à l'infini et donc des lignes épipolaires parallèles.

Soit la matrice de projection perspective $\mathbf{M}_L = \mathbf{K}_L \begin{bmatrix}\mathbf{R}_L & \mathbf{t}_L\end{bmatrix}$ associée à la caméra gauche, et la matrice $\mathbf{M}_R = \mathbf{K}_R \begin{bmatrix}\mathbf{R}_R & \mathbf{t}_R\end{bmatrix}$ associée à la caméra droite. Il est possible de définir deux nouvelles matrices de projection tel que les centres optiques des deux caméras $\mathbf{c}_L$ et $\mathbf{c}_R$ restent inchangés, représentés sur la figure 3.8, définis tels que :

$$\mathbf{M}_L^n = \mathbf{K^n} \begin{bmatrix}\mathbf{R} & -\mathbf{R}\mathbf{c_L}\end{bmatrix} \quad \mathbf{M}_R^n = \mathbf{K^n} \begin{bmatrix}\mathbf{R} & -\mathbf{R}\mathbf{c_R}\end{bmatrix}, \tag{3.12}$$

où la matrice de rotation $\mathbf{R}$ commune au deux nouvelles matrices de projection est définie par :

$$\mathbf{R} = \begin{bmatrix}\mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T\end{bmatrix}. \tag{3.13}$$

Les vecteurs lignes de la matrice $\mathbf{R}$ sont définis tels que le nouvel axe X^n soit parallèle à la *baseline* :

$$\mathbf{r}_1 = (\mathbf{c}_L - \mathbf{c}_R) / \|\mathbf{c}_L - \mathbf{c}_R\|, \tag{3.14}$$

le nouvel axe Y^n soit orthogonal à X^n :

$$\mathbf{r}_2 = \mathbf{k} \wedge \mathbf{r}_1, \tag{3.15}$$

où le vecteur unitaire $\mathbf{k}$ fixe la position du nouvel axe Y^n et peut être choisi arbitrairement. En pratique, comme spécifié dans Fusiello *et al.* (2000), le vecteur $\mathbf{k}$ est choisi selon l'ancien axe Z de la caméra gauche. Le nouvel axe Z^n est définit orthogonal à X^n et Y^n :

$$\mathbf{r}_3 = \mathbf{r}_1 \wedge \mathbf{r}_2. \tag{3.16}$$

A partir de la nouvelle matrice de rotation $\mathbf{R}$ et de la nouvelle matrice des paramètres intrinsèques $\mathbf{K}^n$ (*i.e.* $\mathbf{K}^n = \mathbf{K}_L$), il est possible de calculer les matrices d'homographie $\mathbf{H}_L$ et $\mathbf{H}_R$ par :

$$\mathbf{H}_L = \mathbf{K}^n\mathbf{R}(\mathbf{K}_L\mathbf{R}_L)^{-1}, \quad \mathbf{H}_R = \mathbf{K}^n\mathbf{R}(\mathbf{K}_R\mathbf{R}_R)^{-1}. \tag{3.17}$$

Les images originales gauches et droites sont alors rectifiées par une transformation de *warping* :

$$\mathcal{I}_L^r(\mathbf{p}_L) = \mathcal{I}_L(w(\mathbf{H}_L, \mathbf{p}_L)), \quad \mathcal{I}_R^r(\mathbf{p}_R) = \mathcal{I}_L(w(\mathbf{H}_R, \mathbf{p}_R)), \tag{3.18}$$

où la fonction $w(.)$ transforme les points $\overline{\mathbf{p}}$ en coordonnées homogènes par l'homographie $\mathbf{H}$ tel que :

$$\overline{\mathbf{p}}^w = \frac{\mathbf{H}\overline{\mathbf{p}}}{\mathbf{e}_3^T\mathbf{H}\overline{\mathbf{p}}}. \tag{3.19}$$

Comme pour les transformations de *warping* perspectives présentées en section 2.4.1, les valeurs des nouvelles intensités aux pixels $\mathbf{p}^w$ sont obtenues par interpolation bilinéaire.

La nouvelle matrice des paramètres extrinsèques entre la caméra gauche et la caméra droite $\mathbf{T}^N$ est alors une translation pure (correspondant à la distance des centres optiques $\mathbf{c_L}$ et $\mathbf{c_R}$) suivant l'axe X^n de la nouvelle caméra gauche :

$$\mathbf{T}^N = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

La figure 3.9 montre une paire d'images stéréo du système avant et après rectification ainsi que la carte de disparité associée à la caméra gauche (*cf.* section 3.2.3.2). Bien, que seulement la région centrale soit mise en correspondance dans cet exemple, les régions situées sur les bords gauche et droit des images rectifiées sont en fait en recouvrement avec d'autres caméras du système et donc mises en correspondance dans un autre espace de rectification.

3.2.3.2 Mise en correspondance dense

A partir d'une paire d'images stéréo rectifiées, il est possible d'utiliser des techniques classiques de mise en correspondance dense. C'est à dire trouver, pour chaque pixel $\mathbf{p}$ de l'image gauche $\boldsymbol{\mathcal{I}}_L$, le pixel correspondant dans l'image droite $\boldsymbol{\mathcal{I}}_R$ et vice versa, si possible avec une précision sub-pixellique :

$$\mathbf{d} = \arg\min_{\mathbf{d}} \mathcal{C}(\boldsymbol{\mathcal{I}}_L(\mathbf{p}), \boldsymbol{\mathcal{I}}_R(\mathbf{p}-\mathbf{d})), \tag{3.21}$$

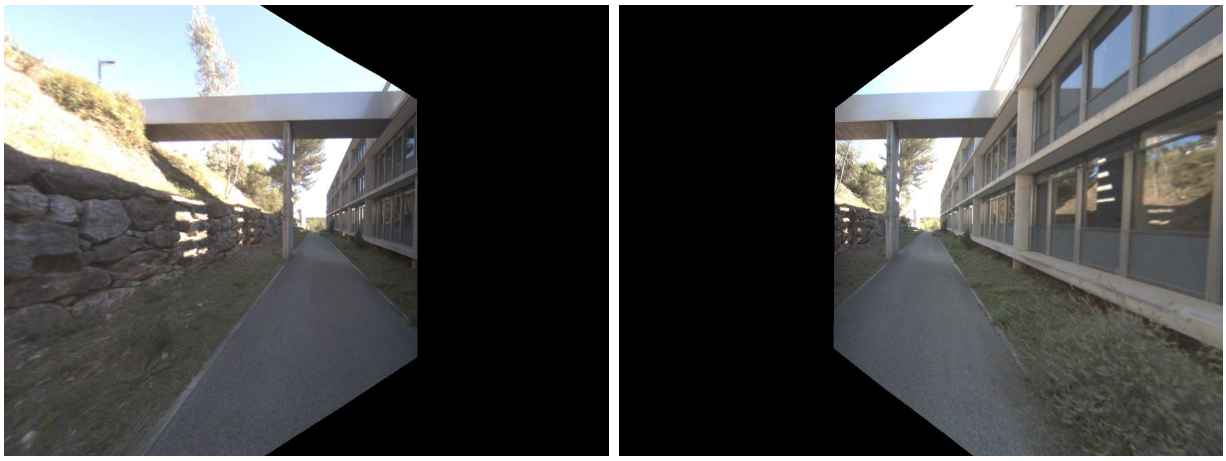
où $\mathbf{d} = (d_x, 0)$ est la disparité. Classiquement, la disparité est obtenue par une recherche exhaustive le long des lignes épipolaires. La fonction de coût $\mathcal{C}$ définit la métrique utilisée pour évaluer l'erreur, ainsi que la dimension des fenêtres de calcul. Par exemple, pour une somme des différences absolues (SAD), la fonction s'écrit :

$$\mathcal{C}(\mathbf{p}, \mathbf{d}) = \sum_{\mathbf{q} \in N_{\mathbf{p}}} |\boldsymbol{\mathcal{I}}_L(\mathbf{q}) - \boldsymbol{\mathcal{I}}_R(\mathbf{q}-\mathbf{d})|, \tag{3.22}$$

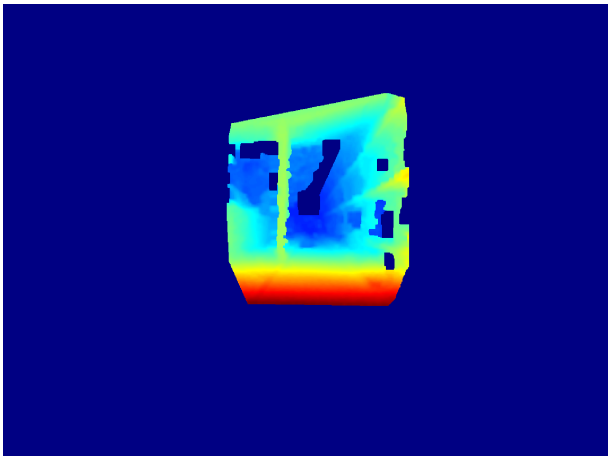
où $N_{\mathbf{p}}$ définit la fenêtre de mise en correspondance. Une grande fenêtre produit en général des cartes disparités moins bruitées mais lissées.

La littérature sur la mise en correspondance dense est abondante : Scharstein & Szeliski (2002) propose un outil de comparaison des techniques. Les différentes approches sont assez variés, allant des méthodes locales, tel que le *Block matching* définit précédemment, à des techniques globales de type *Graph-Cut* (Kolmogorov & Zabih (2001)) ou *Semi-global Block Matching* Hirschmuller (2008) qui permet d'ajouter une contrainte de lissage à la fonction de coût. D'autres méthodes font l'hypothèse de surfaces planaires par morceaux (Ogale & Aloimonos (2005); Geiger *et al.* (2010)), ce qui simplifie la mise en correspondance. Dans Hirschmuller & Scharstein (2009), une comparaison de plusieurs fonctions de coût est effectuée en présence de différences radiométriques.

(a) Images gauche $\mathcal{I}_L$ et droite $\mathcal{I}_R$, avec correction des distorsions, de dimensions 1292×964 pixels.



(b) Images rectifiées gauche $\mathcal{I}_L^r$ et droite $\mathcal{I}_R^r$, par les transformations de l'équation (3.18)



(c) Carte des disparités associées à l'image gauche rectifiée.

FIG. 3.9 – Rectification d'une paire d'images stéréo et carte de disparité associée à la caméra gauche.

Bien que les travaux présentés dans cette thèse ne sont pas orientés sur la mise en correspondance dense, plusieurs difficultés ont étés rencontrées, en partie à cause de la configuration divergente des caméras. En effet, la mise en correspondance dense de vues divergentes n'a pas été beaucoup étudiée dans la littérature, hormis par Tola *et al.* (2010) où des descripteurs (DAISY) sont utilisés pour la mise en correspondance d'images à large *baseline*.

Pour le système présenté dans cette thèse, l'angle formé entre les axes optiques de chaque caméra est clairement divergent (60°). De plus la *baseline* séparant chaque paire stéréo peut être considérée comme large (65 cm). Ces deux contraintes induisent des différences de résolution entre les images stéréo rectifiées non négligeables, et un large domaine de recherche des disparités, synonyme de minima locaux. Plusieurs algorithmes de mise en correspondance dense ont été testés sur les images du système multi-caméras : les techniques standard tel que Ogale & Aloimonos (2005) n'ont pas produit de résultats satisfaisants. En revanche, des méthodes tel que Hirschmuller (2008); Tola *et al.* (2010); Geiger *et al.* (2010) se sont avérées efficaces sur les images du système. Dans la suite de ces travaux, l'algorithme du *Semi-Global Block Matching* Hirschmuller (2008) a été utilisé car il offre un bon compromis entre qualité des cartes de disparité et temps de calcul.

3.2.3.3 Triangulation

Une fois la mise en correspondance dense effectuée sur les 6 paires stéréo du système, un nuage de point 3D peut être extrait par triangulation (Hartley & Sturm (1995)). Dans le cas d'une paire stéréo rectifiée, les paramètres extrinsèques du système correspondent à une translation pure suivant l'axe X de la caméra gauche rectifiée. La profondeur associée au pixel $\mathbf{p}$, exprimée dans le repère de la caméra est directement proportionnelle à l'inverse de la disparité :

$$Z = t_x \frac{f}{d_x}, \tag{3.23}$$

où t_x correspond à la *baseline* de la paire stéréo rectifiée, f est la distance focale de la caméra et d_x est la valeur de la disparité exprimée en pixels au point $\mathbf{p}$. Par conséquent, le point $\mathbf{P} \in \mathbb{R}^3$, associé au pixel $\mathbf{p}$ de l'image est défini par :

$$\mathbf{P} = Z\mathbf{K}^{-1}\overline{\mathbf{p}}. \tag{3.24}$$

La triangulation des disparités des 6 paires d'images stéréo permet de construire un nuage de points 3D.

3.2.4 Fusion de l'information

Le nuage de point 3D obtenu sur chaque paire stéréo est alors re-centré et projeté sur une sphère virtuelle positionnée à l'intérieur du système multi-caméra par une projection sphérique, représentée sur la figure 3.2.4, définie telle que :

$$\mathbf{q}_E = \begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} = \frac{\mathbf{R}_i^c\mathbf{P} + \mathbf{t}_i^c}{\|\mathbf{R}_i^c\mathbf{P} + \mathbf{t}_i^c\|}, \tag{3.25}$$

où la transformation $\mathbf{T}_i^c = \begin{bmatrix} \mathbf{R}_i^c & \mathbf{t}_i^c \end{bmatrix}$ transfère rigidement les points $\mathbf{P}$ vers la sphère virtuelle $\boldsymbol{\mathcal{S}}$ dont la position peut être choisie arbitrairement (*e.g.* centre de gravité des caméras).

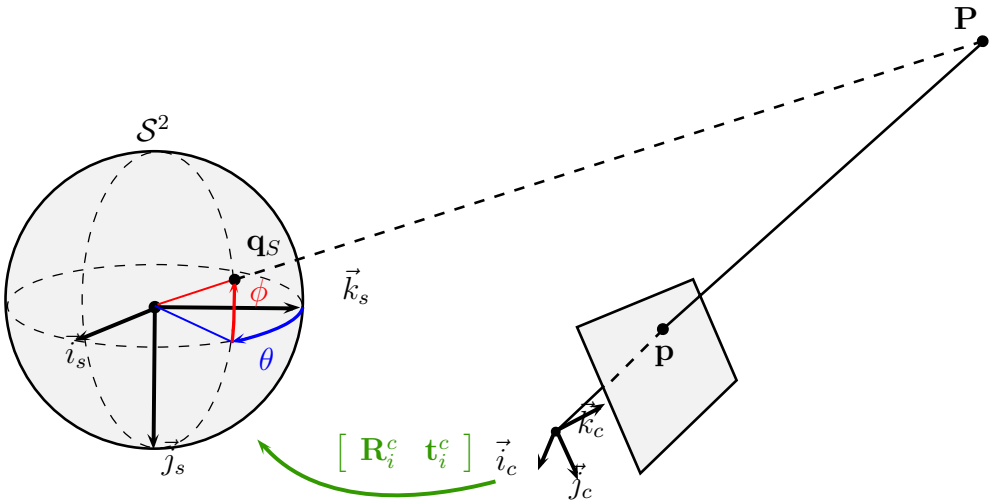


FIG. 3.10 – Projection sphérique.

Les points $\mathbf{q}_E$ en coordonnées cartésiennes sont convertis en coordonnées sphériques par :

$$\mathbf{q}_S = \begin{bmatrix} \theta \\ \phi \\ \rho \end{bmatrix} = \begin{bmatrix} \arctan(z_S/x_S) \\ \arctan(y_S/\sqrt{x_S^2 + z_S^2}) \\ \sqrt{z_S^2 + y_S^2 + z_S^2} \end{bmatrix} \tag{3.26}$$

L'information de profondeur Z_S peut alors être interpolée aux valeurs d'échantillonnage de la sphère $\boldsymbol{\mathcal{P}}_S$ (défini en section 3.2.5), ce qui permet d'obtenir la carte de profondeur $\boldsymbol{\mathcal{Z}}_S$ de la sphère augmentée.

L'étape suivante consiste à ramener l'information photométrique des images sur la sphère. Puisque la carte de profondeur est disponible, les intensités des 6 images originales sont transférées par une fonction de *warping* 3D vers la sphère virtuelle par :

$$\boldsymbol{\mathcal{I}}_S(\mathbf{q}_S) = \boldsymbol{\alpha}_1 \boldsymbol{\mathcal{I}}_1\Big(w(\mathbf{K}_1, \mathbf{T}_1^c, Z_S, \mathbf{q}_S)\Big) + \ldots + \boldsymbol{\alpha}_6 \boldsymbol{\mathcal{I}}_6\Big(w(\mathbf{K}_6, \mathbf{T}_6^c, Z_S, \mathbf{q}_S)\Big). \tag{3.27}$$

Cette fonction de *warping* est similaire à celle du *mosaicing* classique de l'équation (3.6). Cependant, l'information de distance Z_S contenue dans la carte de profondeur $\boldsymbol{\mathcal{Z}}_S$ permet de ne pas négliger les translations des caméras, et donc d'effectuer une transformation 3D par les poses $\mathbf{T}_i^c$, ce qui permet de transférer correctement les intensités des images vers la sphère. Les cœfficients $\boldsymbol{\alpha}_i$, permettent de fusionner les intensités sur les *overlap* entre les images. La fusion la plus basique consiste à prendre la moyenne des intensités.

Cependant, puisque l'information photométrique provient de plusieurs caméras, ayant des paramètres d'exposition ou d'ouverture différents, des discontinuités colorimétriques peuvent apparaitre entre les régions en recouvrement. Ce problème, bien connu dans le domaine du *mosaicing* d'images Szeliski (2006) peut être résolu en utilisant une fonction de fusion (*blending*). Dans notre cas, la fonction de fusion proposée par Burt & Adelson (1983) peut être utilisée : le *Laplacien Blending*. Au lieu d'utiliser une simple moyenne des intensités sur la bande de recouvrement, la largeur de la bande est adaptée en fonction d'une pyramide passe-bande (pyramide Laplacienne) construite pour chaque image. Cette pyramide est une décomposition fréquentielle des intensités de l'image, réversible, et ne dégradant pas les données. Pour les basses fréquences, une large bande de la région de

recouvrement est utilisée et permet de calculer les masque de poids $\boldsymbol{\alpha}_i$ pour chaque pixel. La bande de fusion utilisée est alors réduite pour chaque niveau, jusqu'à la plus haute fréquence (détails des images). Chaque niveau de la pyramide est alors fusionné en fonction du masque de poids calculé. La transformation inverse de la pyramide Laplacienne est appliquée et permet de reconstruire l'image fusionnée. Grâce à ce type de fusion, les basses fréquences sont moyennées sur tout le recouvrement, et les détails sur une zone plus faible.

Les images de la figure 3.11 montrent des sphères photométriques de 2 Méga-pixels obtenues à la fin du processus de reconstruction. Puisque l'information 3D est utilisée pour re-projeter l'information photométrique, les régions où la mise à en correspondance dense a échouée ne sont pas reconstruites et apparaissent en noir dans les images. Grâce à la fusion des intensités, la colorimétrie est continue sur toute la sphère.

3.2.5 Échantillonnage d'une sphère

3.2.5.1 Échantillonnage à angles constants

La technique classique d'échantillonnage sur la sphère unitaire consiste à échantillonner les points $\boldsymbol{\mathcal{P}}_S = (\theta, \phi)$ à pas constant, avec les angles $\theta \in [-\pi, \pi]$ et $\phi \in [0, \pi]$, par les pas d'échantillonnage $d\theta$ et $d\phi$:

$$d\theta = \frac{2\pi}{m}, \quad d\phi = \frac{\pi}{n}, \tag{3.28}$$

où m désigne le nombre d'échantillons en latitude et n le nombre d'échantillons en longitude. Avec un tel échantillonnage la distribution des points échantillonnés sur la sphère n'est pas uniforme : les pôles sont sur-échantillonnés (*cf.* Fig. 3.12(a)). Cela peut poser quelques problèmes :

- Des artéfacts liés au sur-échantillonnage apparaissent sur les images reconstruites (*aliasing*).
- Un nombre de points trop important dans une direction peut biaiser l'estimation de pose et favoriser l'observation d'un degré de liberté au détriment des autres directions.

Dans certains cas, il peut être nécessaire d'utiliser un échantillonnage le plus uniforme possible pour conserver des données consistantes sur toute la sphère.

3.2.5.2 Méthodes d'échantillonnage uniforme

Dans la littérature, plusieurs techniques s'approchent d'un échantillonnage quasi uniforme, c'est à dire une distribution des pixels uniforme sur toute la sphère. Dans Saff & Kuijlaars (1997), les points sont distribués sous la forme d'une spirale, (*cf.* figure 3.12(e)). D'autres méthodes se basent sur une subdivision de la sphère en formes géodésiques de type octaèdres ou icosaèdres (*cf.* Tegmark (1996) figures 3.12(c) et 3.12(d)), ou bien en grille triangulaire (Szalay & Brunner (1999)). La méthode du *QuadCube*, projete les points contenus sur les six faces d'un cube sur une sphère (*cf.* figure 3.12(b)). Gorski *et al.* (2005) propose une distribution hiérarchique des pixels de la sphère obtenue de manière analytique (Healpix, *cf.* figure 3.12(f)). Contrairement aux autres méthodes, cette technique possède certaines propriétés adaptées au traitement d'images telles que :

- Une structure hiérarchique permettant d'utiliser des techniques multi-résolution.
- Un voisinage local bien défini.
- Une surface sur la sphère égale pour chaque pixel.

3.2.5.3 Conclusions sur l'échantillonnage

La technique d'échantillonnage Healpix, Gorski *et al.* (2005), a été implémentée et validée hors ligne sur des données de simulation. Cependant plusieurs outils classiques tels que les opérateurs de gradient, l'accès séquentiel aux pixels des images ou simplement les fonctions d'affichage ont besoin d'être redéfinis et optimisés, car contrairement à une image classique, la distribution des pixels n'est pas une grille uniforme mais est stockée dans un structure arborescente. Les données des images panoramiques reconstruites avec le système multi-caméras sont essentiellement situées autour de l'équateur des sphères (*cf.* figure 3.11), là où les distorsions liées au sur-échantillonnage sont les moins importantes. Dans l'implémentation temps réel des algorithmes de localisation en ligne, l'échantillonnage à pas constant a été utilisé afin de privilégier le temps de calcul.

3.3 Conclusion

Ce chapitre a introduit un nouveau capteur permettant l'acquisition d'images sphériques augmentées par une information dense de profondeur. Le système multi-caméras proposé est nouveau dans le sens où contrairement aux techniques classiques, les centres optiques des caméras sont éloignées les uns des autres afin de générer de la disparité entre chaque paire d'images. Cette disparité permet d'utiliser des techniques de mise en correspondance dense, pour obtenir la profondeur. Finalement cette profondeur permet de re-projeter et de fusionner les intensités des images sur une sphère virtuelle. Une technique d'étalonnage du système a également été proposée, et permet de calibrer le système avec une mire classique, en profitant de la fermeture de boucle de la configuration circulaire du système. Cela évite les contraintes d'utilisation d'une mire d'étalonnage englobant tout le système.

Cette nouvelle configuration a également permis d'identifier plusieurs problèmes, liés à la géométrie du système. En effet, la mise correspondance d'images grand angle avec des angles de vue fortement divergents, est délicate et mériterait une étude approfondie.

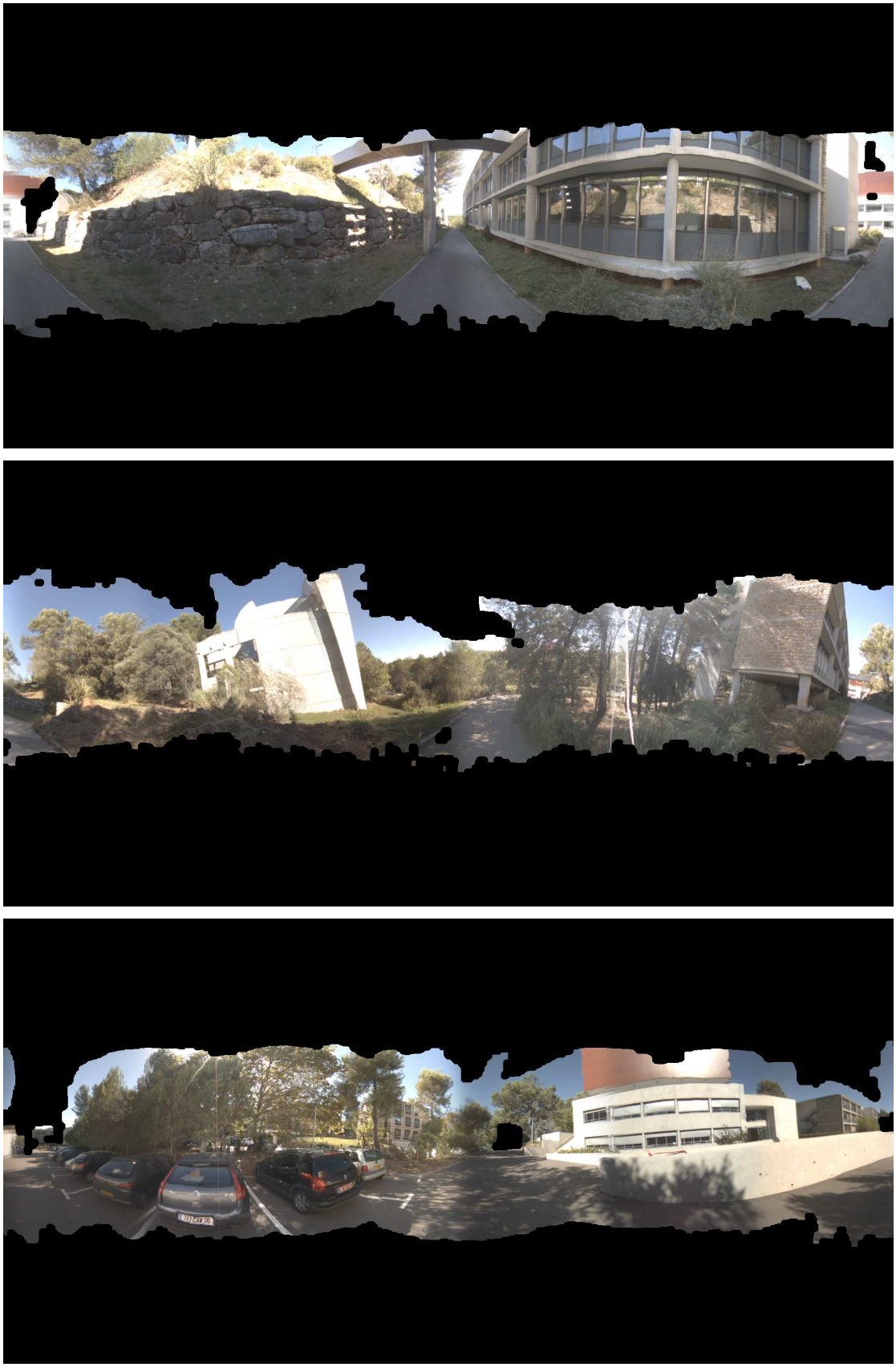
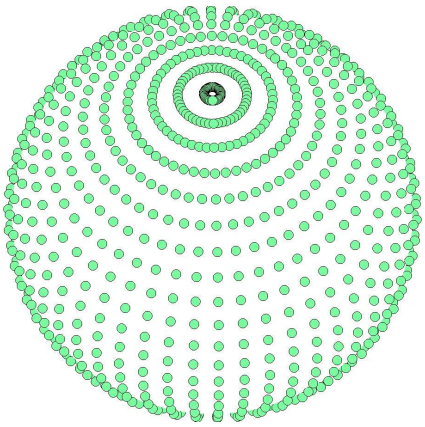
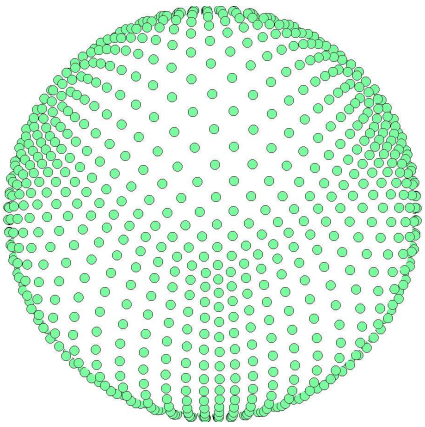


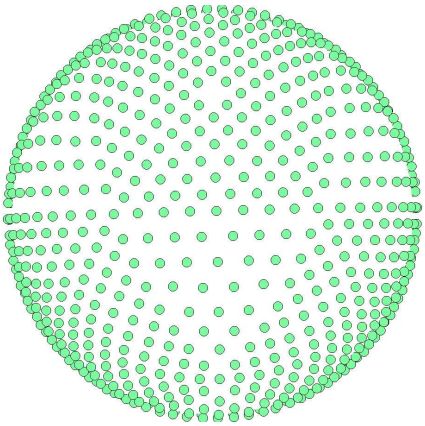
Fig. 3.11 – Images sphériques reconstruites avec le système à *baseline*. Les zones en noir n'ont pas été reconstruites par manque d'information de profondeur.



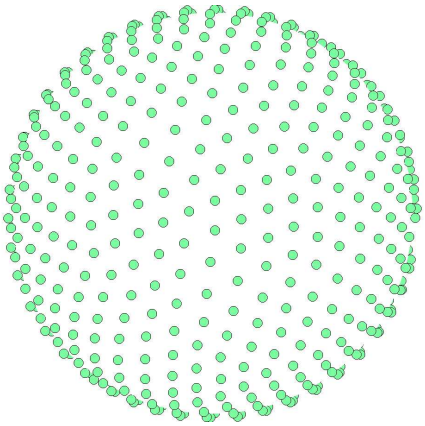
(a) Constant 1250 pixels.



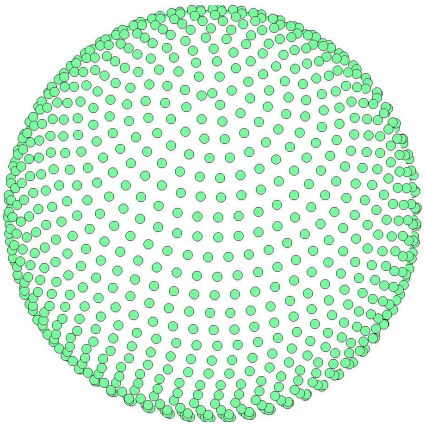
(b) *QuadCube, 1350 pixels.*



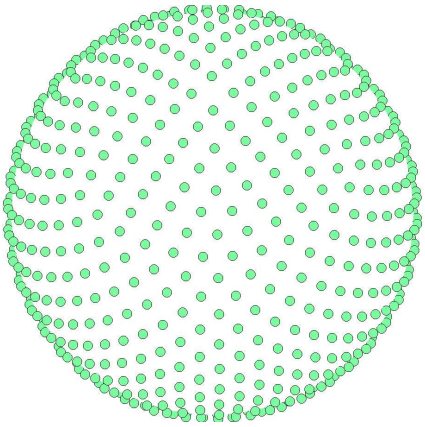
(c) *Octaèdre 2048 pixels.*



(d) *Icosaèdre 1280 pixels.*



(e) Spirale 1280 pixels.



(f) *Healpix 768 pixels.*

FIG. 3.12 – Échantillonnage d'une sphère unitaire. La méthode à pas constant (a) sur-échantillonne les pôles. La méthode *quadcube* (b) n'est pas uniforme et présente des discontinuités. Pour les méthodes, spirale (e), icosaèdre (d) et octaèdre (c), le voisinage entre les pixels est mal défini. La méthode Healpix (f) propose un voisinage bien défini et une structure hiérarchique idéale pour le traitement d'image.

Chapitre 4

Cartographie

4.1 Introduction

Dans le chapitre précédent, un nouveau capteur sphérique a été présenté. Ce capteur permet la construction de sphères visuelles denses augmentées par la profondeur : c'est à dire la représentation locale du modèle. Dans ce chapitre, la construction du graphe global est abordée. Cette étape consiste à estimer la pose 3D reliant les sphères : les arêtes du graphe. Puisque le modèle est destiné à la localisation d'une caméra embarquée sur un véhicule autonome, la pose 3D entre chaque nœud du graphe doit être la plus précise possible. Les solutions classiques de type GPS ne fournissent pas une localisation suffisamment précise ni d'information sur la rotation 3D du système. De plus ces capteurs sont très sensibles aux occultations des bâtiments. Pour résoudre ce problème, une technique directe d'odométrie visuelle exploitant directement les avantages des sphères augmentées est utilisée. Un critère robuste permet de sélectionner automatiquement les sphères constituant le graphe.

4.2 Odométrie visuelle sphérique 3D

4.2.1 Modélisation du problème

Considérant qu'une sphère visuelle augmentée $\boldsymbol{\mathcal{S}}^*$ a été reconstruite à l'instant $t-n$, l'objectif est d'estimer le déplacement relatif du système de caméras à l'instant t par rapport à la sphère de référence $\boldsymbol{\mathcal{S}}^*$. Au lieu d'estimer la pose entre deux sphères successivement reconstruites, il est possible d'estimer directement la pose entre l'ensemble des images perspectives perçues à l'instant t et la sphère de référence reconstruite à l'instant $t-n$. Cela permet d'éviter de reconstruire "inutilement" une sphère, et surtout d'utiliser directement l'information originale des images afin de conserver un maximum de précision dans l'estimation du mouvement.

De la même manière que dans le chapitre 2.6, l'estimation de la pose peut être formulée comme un problème d'optimisation dont l'objectif est de minimiser directement les erreurs d'intensité entre la sphère de référence $\boldsymbol{\mathcal{S}}^*$ et l'ensemble des images perspectives du système $\{\boldsymbol{\mathcal{I}}_1, \boldsymbol{\mathcal{I}}_2, \ldots, \boldsymbol{\mathcal{I}}_6\}$ transférées sur la sphère.

La fonction $w(\mathbf{T}(\tilde{\mathbf{x}}_i); \mathbf{K}(\xi_\mathbf{i}), Z_S, \mathbf{q}_S)$ transfère le point $\mathbf{q}_S$ de la sphère de référence en coordonnées pixelliques $\mathbf{p}_i$ dans l'image $\boldsymbol{\mathcal{I}}_i$ tel que :

$$\mathbf{p}_i = w(\mathbf{T}(\tilde{\mathbf{x}}_i); \mathbf{K}(\xi_\mathbf{i}), Z_S, \mathbf{q}_S) \tag{4.1}$$

où $\mathbf{T}(\tilde{\mathbf{x}}_i)$ correspond au mouvement rigide de la caméra i par rapport à la sphère et $\mathbf{K}(\xi_\mathbf{i})$ correspond aux paramètres intrinsèques de la caméra. Le point $\mathbf{q}_S = (\theta, \phi, Z_S)$ est transformé en coordonnées cartésiennes par :

$$\mathbf{q}_E = Z_S \begin{bmatrix} \cos\theta\cos\phi \\ \sin\phi \\ \sin\theta\cos\phi \end{bmatrix}, \tag{4.2}$$

puis projeté par une projection perspective dans l'image :

$$\overline{\mathbf{p}_i} = \frac{\mathbf{K}(\xi_\mathbf{i}) \begin{bmatrix} \mathbf{R}(\tilde{\mathbf{x}}_\mathbf{i}) & \mathbf{t}(\tilde{\mathbf{x}}_i) \end{bmatrix} \mathbf{q}_E}{\mathbf{e}_3^t \mathbf{K}(\xi_\mathbf{i}) \begin{bmatrix} \mathbf{R}(\tilde{\mathbf{x}}_\mathbf{i}) & \mathbf{t}(\tilde{\mathbf{x}}_i) \end{bmatrix} \mathbf{q}_E}. \tag{4.3}$$

Il est donc possible de définir la fonction de *warping* perspective entre la caméra i et la sphère $\boldsymbol{\mathcal{S}}^*$ par :

$$\boldsymbol{\mathcal{I}}_S^*(\mathbf{q}_S) = \boldsymbol{\mathcal{I}}_i(w(\mathbf{T}(\tilde{\mathbf{x}}_i); \mathbf{K}(\xi_\mathbf{i}), Z_S, \mathbf{q}_S)). \tag{4.4}$$

Puisque le système est calibré, la pose $\mathbf{T}(\tilde{\mathbf{x}}_i)$ peut être exprimée en une composition de deux poses :

$$\mathbf{T}(\tilde{\mathbf{x}}_i) = \mathbf{T}(\mathbf{x}_i^c)\mathbf{T}(\tilde{\mathbf{x}}). \tag{4.5}$$

où $\mathbf{T}(\mathbf{x}_i^c)$ est la pose de la caméra par rapport à la sphère virtuelle courante $\boldsymbol{\mathcal{S}}$ (non reconstruite) et la pose $\mathbf{T}(\tilde{\mathbf{x}})$ représente la pose entre la sphère courante et la sphère de référence (*cf.* figure 4.1).

Dans ce cas, la matrice $\mathbf{T}(\mathbf{x}_i^c)$ est constante et ne dépend que des paramètres extrinsèques de la caméra i obtenus lors de l'étalonnage du système.

Si l'on considère que seulement une approximation $\widehat{\mathbf{T}}$ de $\mathbf{T}(\tilde{\mathbf{x}})$ est disponible, l'objectif est de trouver la transformation incrémentale satisfaisant :

$$\mathbf{T}(\tilde{\mathbf{x}}) = \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}). \tag{4.6}$$

Il est alors possible de définir une fonction d'erreur entre les intensités de la caméra i et celle de la sphère $\boldsymbol{\mathcal{S}}^*$, fonction de l'inconnue $\mathbf{x}$:

$$\mathbf{e}_i(\mathbf{x}) = \boldsymbol{\mathcal{I}}_i(w(\mathbf{T}(\mathbf{x}_i^c)\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); \mathbf{K}(\xi_\mathbf{i}), Z_S, \mathbf{q}_S)) - \boldsymbol{\mathcal{I}}_S^*(\mathbf{q}_S) \tag{4.7}$$

4.2.2 Minimisation globale

En suivant la formulation de l'erreur précédente, il est possible de définir un vecteur d'erreur global en empilant les 6 vecteurs d'erreur $\mathbf{e}_i(\mathbf{x})$ issus des 6 cameras :

$$\mathbf{e}(\mathbf{x}) = \begin{bmatrix} \mathbf{e}_i(\mathbf{x}) & \mathbf{e}_2(\mathbf{x}) & \dots & \mathbf{e}_6(\mathbf{x}) \end{bmatrix}^T \tag{4.8}$$

Cette fonction d'erreur permet de définir la fonction de coût robuste suivante :

$$\mathcal{O}(\mathbf{x}) = \rho(\mathbf{e}(\mathbf{x})), \tag{4.9}$$

où ρ est la fonction robuste définie dans le chapitre 2. Cette erreur peut être minimisée itérativement de manière efficace, avec la méthode directe définie en section 2.6. La pose $\widehat{\mathbf{T}}$ obtenue après recalage des images correspond au déplacement entre la sphère de référence $\boldsymbol{\mathcal{S}}^*$ et la sphère courante $\boldsymbol{\mathcal{S}}$ et permet d'initialiser la minimisation à l'instant $t+1$.

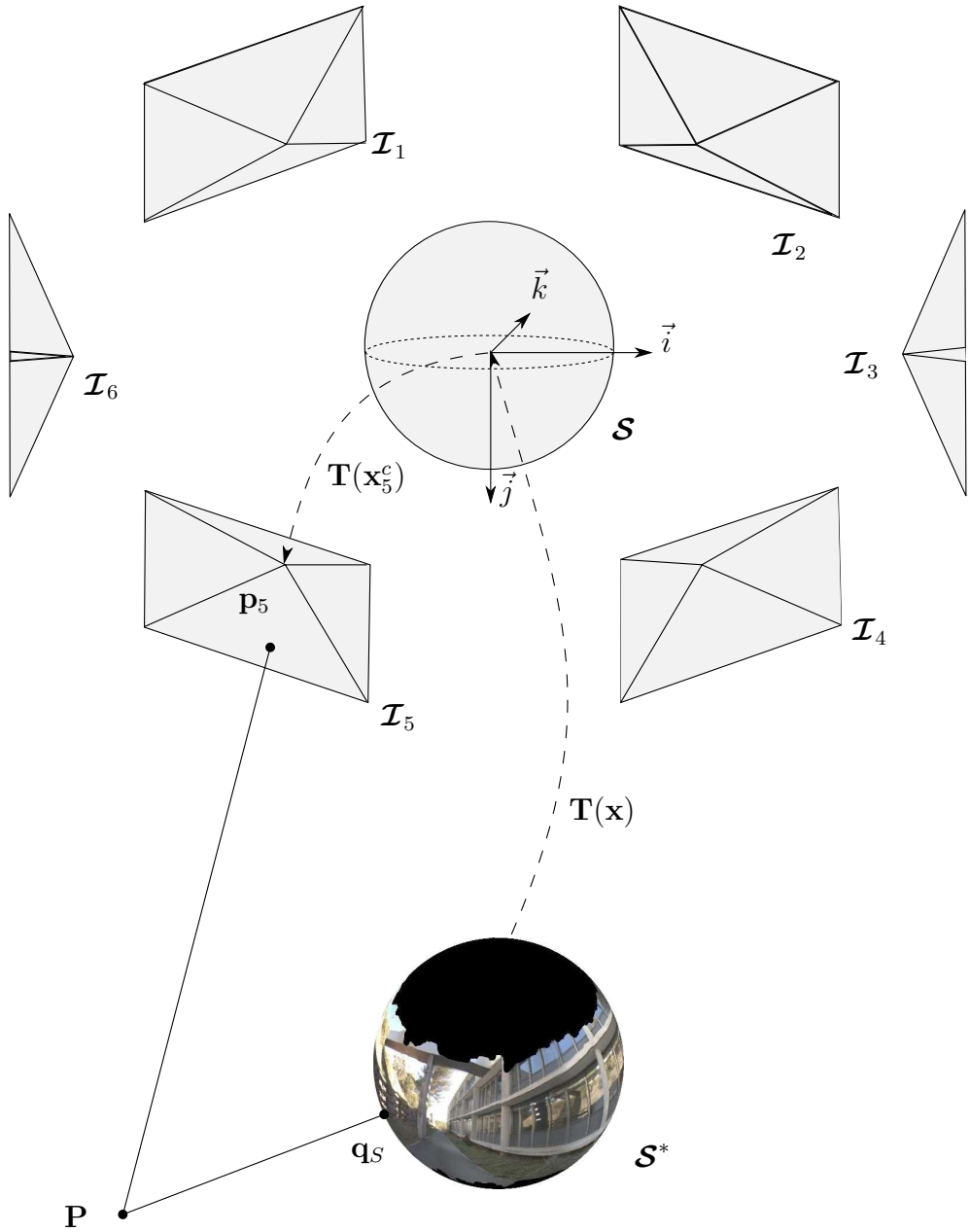


FIG. 4.1 – Odométrie visuelle sphérique 3D. Le mouvement $\mathbf{T}(\mathbf{x})$ entre la sphère de référence et la sphère courante est estimé en transformant directement les 6 images courantes sur la sphère de référence.

4.3 Sélection automatique des sphères du graphe

Le graphe de sphères augmentées, doit être évidemment le moins redondant possible. Dans ce cas il n'est pas nécessaire de reconstruire chaque sphère issue de la séquence d'image (enregistrée à 30 Hz). Une stratégie couramment utilisée en localisation visuelle est de conserver le plus longtemps possible la sphère de référence $\boldsymbol{\mathcal{S}}^*$ dans la minimisation, et de choisir d'ajouter une nouvelle sphère au graphe en fonction d'un critère robuste. La mesure choisie dans ces travaux est une mesure basée image : la valeur absolue des écarts à la médiane (MAD, *cf.* détails dans la section 2.6.5), qui représente l'échelle de la distribution de l'erreur. Typiquement une nouvelle sphère est reconstruite, ajoutée au graphe et utilisée comme référence lorsque la valeur de la MAD, pour la valeur estimée de $\mathbf{x} = \widehat{\mathbf{x}}$, dépasse un seuil λ_1 :

$$\lambda_1 < median(|\mathbf{e}(\widehat{\mathbf{x}}) - median(\mathbf{e}(\widehat{\mathbf{x}}))|). \tag{4.10}$$

Ce critère exprime dans quelle mesure la photométrie de la scène a changée entre les images courantes et la sphère de référence. Cette variation photométrique est directement liée aux mouvements géométriques de la scène via la fonction d'erreur (4.7) et permet de quantifier :

- La quantité d'occultations présentes entre les images (liée à la géométrie et aux points de vue).
- Les changements de résolution présents entre la sphère de référence et les images (*i.e.* la distance entre les points de vue).

Puisqu'une sphère couvre un champ visuel à 360°, la sphère de référence et les images courantes sont en *overlap* permanent. Dans le cas où un capteur à champ visuel limité est utilisé pour reconstruire la base de donnée (*i.e.* paire stéréo classique), il faut prendre en compte le nombre de points 3D visibles entre les images courantes et les images de référence. Dans ce cas, un deuxième seuil doit être utilisé :

$$\lambda_2 > N_{warp}(\widehat{\mathbf{x}}), \tag{4.11}$$

où $N_{warp}(\widehat{\mathbf{x}})$ est le nombre de points visibles entre les images courantes et l'image de référence après alignement, et λ_2 est le nombre de points minimal pré-définit.

Puisque la technique de localisation est de type odométrie visuelle, l'estimation de la pose consiste à intégrer les déplacements inter-images le long de la séquence d'apprentissage, ainsi que les erreurs de toute la chaine de reconstruction des sphères (*i.e.* mise en correspondance dense, re-projection *etc*), ce qui génère des erreurs de dérive. Le fait de ne pas reconstruire systématiquement une sphère réduit la quantité d'erreurs introduites dans la boucle d'estimation et donc la dérive totale de la trajectoire estimée.

La figure 4.2, montre l'évolution de la valeur de la MAD le long d'une trajectoire de 500 images, en fonction de la distance parcourue Δ par le véhicule. Dans les régions où des bâtiments sont loin des caméras ($0 < \Delta < 8$ et $16 < \Delta < 25$), la MAD croit lentement en fonction de la distance parcourue. Dans les régions où les bâtiments sont proches des caméras ($8 < \Delta < 16$), la valeur de la MAD croit plus rapidement et atteint le seuil λ plus fréquemment (ici fixé à 5 % de la valeur maximale des intensités des images, *i.e.* 255 niveaux de gris), ajoutant plus de sphères de référence au graphe.

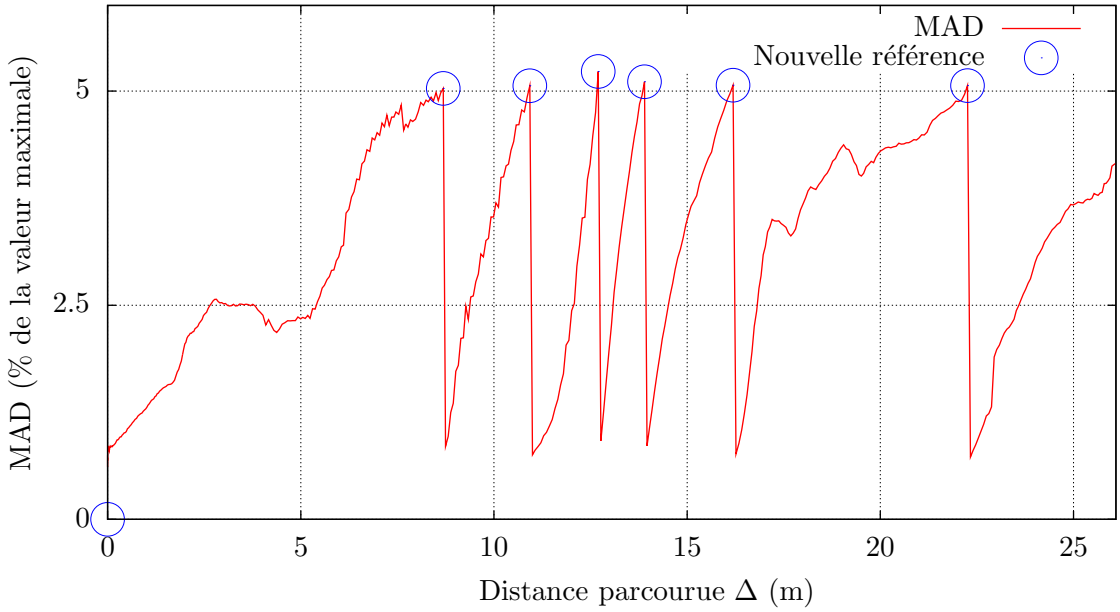


FIG. 4.2 – Évolution de la MAD en fonction de la distance parcourue. En fonction de la configuration de la scène, la valeur de la MAD croit plus ou moins rapidement. Une nouvelle sphère de référence est construite lorsque la MAD dépasse 5 % de la valeur maximale des intensités des images.

4.4 Résultats expérimentaux

4.4.1 Positionnement des sphères

Une séquence de 7364×6 images à été acquise sur le site de INRIA Sophia Antipolis à l'aide du système sphérique embarqué sur un véhicule de type Cycab. La trajectoire parcourue par le véhicule est d'environ 1.5 km et représente des environnements très variés, contenant des zones dégagées avec des bâtiments éloignés, des corridors, des véhicules stationnés, de la végétation, des virages serrés, du dénivelé, le tout nécessitant une estimation robuste et précise des 6 degrés de liberté du véhicule. La phase de construction des sphères et le positionnement a été calculé hors ligne à une fréquence de calcul d'environ 1 Hz.

Puisque la technique utilisée est de type odométrie visuelle, la dérive intégrée le long de la séquence (en général $\leq 1\%$) peut entrainer des graphes globalement inconsistants, ou redondant (*i.e.* une route cartographié deux fois dans des directions opposées). Pour corriger la dérive et supprimer les sphères redondantes, la technique de détection de fermetures de boucle sphérique proposée par Chapoulie *et al.* (2011) a été utilisée. Cette méthode utilise des descripteurs *SIFT*, dont la distribution sur la sphère est représentée par des histogrammes. Un dictionnaire construit incrémentalement en ligne, permet de détecter les images dont l'apparence est similaire, quelque soit l'orientation de l'image, ce qui est particulièrement bien adapté au graphe d'images sphériques augmentées.

Les fermetures de boucles détectées ont permis d'ajouter de nouvelles arêtes à l'intérieur du graphe. Le graphe final contenant de nouvelles contraintes a alors été optimisé à l'aide de la librairie *TORO* (Grisetti *et al.* (2009)), qui est une implémentation de la méthode de Grisetti *et al.* (2007). Cet algorithme est une extension de la méthode proposée par Olson *et al.* (2006), et permet de minimiser les erreurs introduites par les contraintes du graphe,

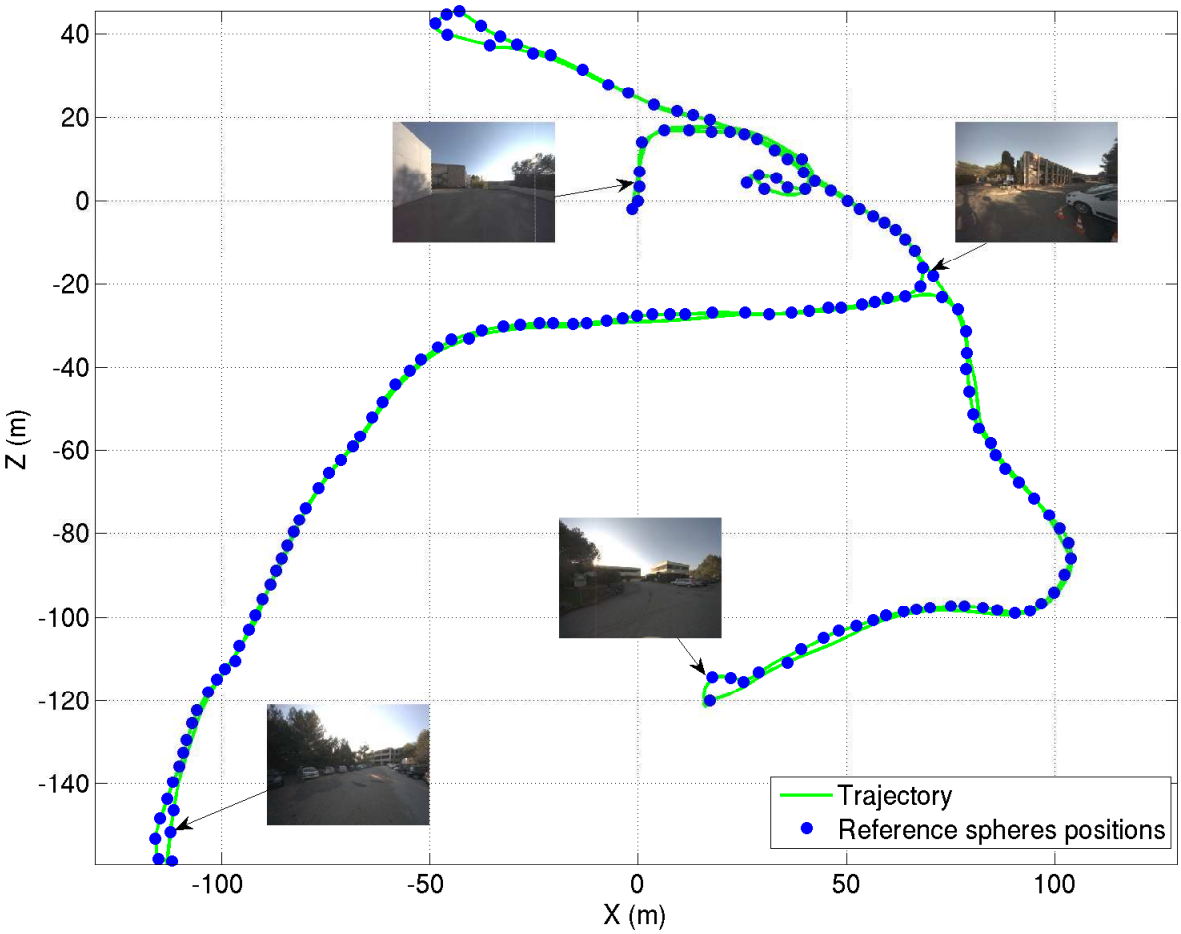


Fig. 4.3 – Graphe de sphères couvrant 1.5 kms.

par une méthode de descente de gradient stochastique. Après optimisation et suppression des sphères détectées par l'algorithme de fermeture de boucle, le graphe final contient 310 sphères augmentées.

La sélection des sphères de référence et la détection de fermetures de boucles ont permis de compresser les 7364 images initiales en 310 sphères de référence. La figure 4.3 montre la trajectoire obtenue après détection de fermetures de boucles et optimisation du graphe, ainsi que certaines images clés de la trajectoire.

4.4.2 Navigation virtuelle photo-réaliste

La représentation sphérique finale, peut être utilisée pour synthétiser des images virtuelles photo-réalistes, à l'aide d'une technique similaire à celles développées en rendu basé image (*image-based rendering*, *cf.* Gortler *et al.* (1996); Levoy & Hanrahan (1996); Debevec *et al.* (1996)). Une application temps-réel a été réalisée avec la librairie de rendu graphique OpenGL, et permet une navigation virtuelle réaliste à l'intérieur du graphe. Le principe illustre une utilisation alternative de la base de donnée sphérique, basée sur le même principe que la localisation en ligne qui sera détaillée dans le chapitre suivant. Une caméra virtuelle est simplement déplacée manuellement par l'utilisateur (clavier et souris), à l'intérieur du graphe. La sphère visuelle augmentée la plus proche de la caméra est alors utilisée

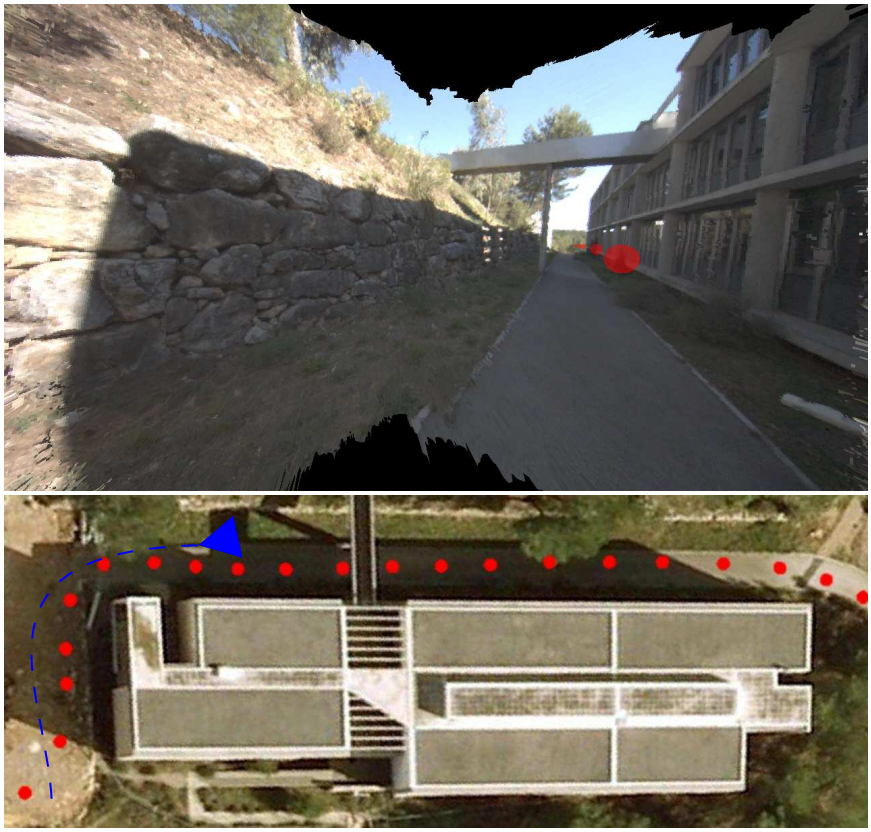


FIG. 4.4 – Haut : Image de synthèse générée à partir d'une sphère augmentée. Bas : Vue aérienne illustrant l'utilisation virtuelle du graphe.

pour générer une vue de synthèse à la position de la caméra virtuelle. La figure 4.4 montre une image virtuelle rendue à l'aide d'une sphère augmentée. On peut voir que localement autour du graphe, il est possible de générer des images de synthèse photo-réalistes. De plus, contrairement aux approches 2D telles celles utilisées dans Google Street View, la transition entre deux sphères est visuellement réaliste, grâce à l'information 3D contenue dans les images et à la précision de l'estimation des arêtes du graphe (poses inter-sphères).

4.5 Conclusion

Dans ce chapitre, le positionnement précis des sphères visuelles augmentées a été présenté. Une technique directe de localisation visuelle est utilisée pour estimer le déplacement des caméras du capteur sphérique le long d'une trajectoire d'apprentissage. L'emploi d'un capteur sphérique contraint très bien l'estimation des mouvements 3D, de plus tous les pixels des images sont utilisés dans la minimisation, ce qui permet une estimation robuste et précise de la pose des nœuds du graphe. Un critère robuste est utilisé pour ajouter une sphère au graphe d'apprentissage. Ce critère basé sur une erreur photométrique prend en compte les changements géométriques de la scène (*i.e.* occultations). Cette sélection permet de compresser les images de la séquence d'apprentissage à quelques images sphériques clés. Dans les résultats expérimentaux, le système a cartographié de manière automatique de larges environnements urbains. Finalement, une utilisation alternative du graphe d'images sphériques est proposée pour une application immersive de navigation virtuelle

photo-réaliste, illustrant l'utilisation en ligne du graphe.

Pour la localisation, il a été choisi de ne pas effectuer d'ajustement de faisceau sur des fenêtres de visibilité, car cela nécessite de reconstruire chaque sphère de la séquence, ce qui est d'une part coûteux en temps de calcul, et d'autre part n'a pas montré de réel gain sur le positionnement des sphères. Dans l'idéal, il serait intéressant de coupler le système de vision à un GPS différentiel pour la construction de la base de donnée, ce qui permettrait de géo-référencer les sphères, et d'utiliser un algorithme d'optimisation tel que Kummerle *et al.* (2011) pour corriger efficacement la dérive et conserver une consistance globale sur les trajectoires estimées.

Chapitre 5

Sélection d'information

5.1 Introduction

Dans le cadre d'une application de navigation autonome, il est important de garder à l'esprit la notion de temps-réel : c'est à dire effectuer les calculs nécessaires à la localisation à la fréquence de la caméra, afin d'effectuer le contrôle d'un robot. Le graphe de sphères augmentées construit lors de la phase d'apprentissage est destiné à être utilisé en temps réel, pour recaler l'image perçue par une caméra avec une technique directe d'alignement d'images. La propriété principale des techniques directes est de minimiser une erreur d'intensité entre tous les pixels communs aux images. Cependant, les images panoramiques de la base de données peuvent être de grande résolution. Dans ce cas, le nombre de pixels utilisés lors de la minimisation peut être trop important pour une utilisation en temps-réel : il est indispensable de sélectionner l'information utile, si possible lors de la construction de la base de donnée d'apprentissage, pour éviter d'effectuer les calculs en ligne.

En général, l'information contenue dans une image est extrêmement redondante et une partie n'est pas indispensable pour la localisation. Il est alors possible de réduire la quantité d'information, au minimum nécessaire à la localisation, tout en conservant une bonne robustesse. La première partie de ce chapitre présente un court état de l'art des techniques de sélection d'information. La seconde partie, propose une nouvelle méthode de sélection de pixels. Contrairement aux méthodes classiques, l'algorithme de sélection est directement conçu pour conditionner au mieux la technique de minimisation directe utilisée dans l'estimation de pose. La méthode de sélection prend en compte à la fois le gradient photométrique et le gradient géométrique des données. Cette sélection permet d'effectuer un tri des pixels lors de la phase d'apprentissage. Lors de la phase de localisation en ligne, seulement les meilleurs pixels seront utilisés dans la localisation, ce qui permet d'effectuer les calculs nécessaire au positionnement en temps-réel.

5.2 État de l'art

Dans le domaine du suivi basé point d'intérêts, la sélection d'information occupe une place importante. L'objectif principal étant d'extraire l'information discriminante et stable des images afin de faciliter la mise en correspondance entre deux images. Par exemple, les détecteurs de *corners* tels que Harris & Stephens (1988), Shi & Tomasi (1994), extraient des points saillants, correspondants aux coins dans l'image, en fonction des extrema des gradients photométriques.

Des descripteurs tels que SIFT (Lowe (2004)), SURF (Bay *et al.* (2006)), PCA-SIFT (Ke & Sukthankar (2004)), GLOH (Mikolajczyk & Schmid (2005)) et récemment DAISY (Tola *et al.* (2010)), sont très souvent utilisés en localisation visuelle et recherchent des régions ayant certaines propriétés d'invariance, notamment aux changements d'échelle et aux rotations, afin de faciliter la mise en correspondance.

D'autres approches, essentiellement utilisées pour des environnements intérieurs, recherchent des indices visuels structurants tel que des segments de droites ou des arcs, segmentés par un detecteur de contour Canny (1986), puis extraits par une transformée de Hough Duda & Hart (1971). Hayet (2003) recherche des points d'intérets appartenant à des primitives quadrangulaires, correspondant à des surfaces planaires.

Cependant, les approches décrites précédemment se basent sur l'information photométrique 2D contenue dans les images et ne considèrent pas la structure géométrique 3D globale de la scène contenue dans l'image. De plus elles sont en général utilisées pour extraire une information discriminante, permettant de mettre en correspondance un à un certains pixels entre deux images. Ces techniques ne sont pas adaptées aux méthodes d'alignement d'images directes, qui n'ont pas d'étape de mise en correspondance, le recalage étant effectué itérativement par une technique d'optimisation.

D'une manière plus générale, l'intérêt des pixels d'une image peut être représenté sous la forme d'une carte de saillance, où la valeur associée aux pixels de la carte correspond à un score Itti *et al.* (1998); Kadir & Brady (2001). Le score de chaque pixel peut être obtenu en calculant une entropie locale, une corrélation locale ou encore la dispersion des intensités dans un voisinage.

En ce qui concerne les méthodes directes, le concept de sélection d'information est plutôt orienté sur l'amélioration des performances des techniques d'optimisation :

- Temps de calcul.
- Vitesse de convergence.
- Bassin de convergence.
- Robustesse aux aberrations.
- Précision.

La principale difficulté est de sélectionner un minimum de pixels, tout en conservant de bonnes propriétés de convergence et de robustesse.

Dans Dellaert & Collins (1999), une carte de saillance est construite en triant les pixels en fonction de la variance de leur matrice Jacobienne. Un sous ensemble de pixels est alors obtenu par une sélection aléatoire parmi les meilleurs pixels. La technique est alors appliquée pour du recalage d'images en rotation pure. Dans Benhimane *et al.* (2007), des ensembles de pixels ayant des propriétés de convergence linéaire ou quadratique sont sélectionnés pour du suivi de surfaces planaires, lors d'une phase d'apprentissage. Cette phase consiste à générer des mouvements aléatoires afin de déterminer quels ensembles de pixels contenus dans l'image ont de bonnes propriétés de convergence (*e.g.* convergence en une itération). Cependant cette technique est très dépendante de la phase d'apprentissage et rejette en général des régions à fort gradient photométrique, indispensable à la précision.

En conclusion, dans la littérature, il existe un nombre important de méthodes de sélection de points d'intérêts. Cependant la plupart sont prévues pour conditionner au mieux la mise correspondance locale des pixels, et sont donc inadaptées aux techniques de recalage directes, basées sur une optimisation non linéaire. De plus aucune technique, ne prend en compte réellement l'information géométrique globale de la scène, essentielle à l'observabilité

de certains mouvements. Il est donc nécessaire de définir une méthode de sélection de pixels saillants, prenant en compte l'information photométrique et l'information géométrique de la scène, afin d'accélérer les temps de calculs, sans dégrader la précision ni la robustesse.

5.3 Sélection de pixels saillants

5.3.1 Gradients géométriques et photométriques

Puisque les méthodes directes reposent sur des techniques d'optimisation basées sur le gradient de la fonction d'erreur, les régions non texturées des images n'apportent localement aucune information : par exemple, une région totalement dépourvue de texture locale (mur uniforme) n'est pas utile à la localisation, que ce soit pour un humain ou pour une caméra. En effet, si le gradient photométrique $\nabla_{\mathbf{p}_i}\boldsymbol{\mathcal{I}}(\mathbf{p}_i) = \mathbf{0}$, alors la ligne i correspondante de la matrice Jacobienne $\mathbf{J}_i$, contient seulement des zéros et n'a donc aucune influence sur l'estimation de pose lorsque qu'une méthode du premier ordre est utilisée (*IC* ou *ESM*). Ne pas utiliser ces pixels permet de gagner du temps dans la projection et l'interpolation des pixels lors du *warping*, et dans le calcul de la pseudo-inverse robuste.

Une approche naïve, très souvent utilisée pour accélerer les calculs des méthodes directes, consiste alors à utiliser seulement les meilleurs gradients photométriques des images en effectuant un tri sur leur norme tel que dans Baker & Matthews (2001); Benhimane (2006); Comport *et al.* (2007) :

$$i = \arg\max_i \|\nabla\boldsymbol{\mathcal{I}}(i)\| \tag{5.1}$$

Cependant, ce type de sélection, basé sur une information 2D peut avantager certaines directions de gradient dans l'image, conduisant à une estimation moins précise de certains déplacements. Dans certains cas, il devient même impossible d'observer certaines directions de déplacement, car une où plusieurs colonnes de la matrice Jacobienne contiennent seulement des zéros. Ces cas sont assez courants en environnements extérieurs, on peut citer par exemple :

1. **Translation non-observable** Une image contenant une région fortement texturée, située à l'infini et une région proche faiblement texturée :
   - Si seulement des pixels à l'infini sont utilisés (invariance aux translations pures), il est alors impossible d'estimer des mouvements de translations (*cf.* simulation en section 5.3.2.1).
   - Par exemple, une place (*cf.* figure 5.1), contenant un sol faiblement texturé et des bâtiments situés loin de la caméra.

FIG. 5.1 – Exemple de façades à "l'infini".

2. **Direction non-observable** Une image contenant des droites horizontales ou verticales (parallèles), à forts gradients photométriques :
   - Si seulement ces pixels sont utilisés, seuls les mouvements orthogonaux à ces droites sont observables.
   - Exemple : Une façade contenant des motifs à forts gradients horizontaux, comme sur la figure 5.2.

FIG. 5.2 – Exemple de façade contenant de forts gradients ne contraignant qu'une seule direction du mouvement.

Bien qu'il soit très rare que certains mouvements soient complètement inobservables sur des images réelles, il est très probable qu'une direction de gradient soit sous représentée dans la sélection effectuée, en particulier lorsque très peu de pixels sont utilisés (*i.e.* 5 à 10 % de l'image).

Il est donc nécessaire d'inclure dans la méthode de sélection l'influence de l'information géométrique, indispensable à l'observation de mouvements 3D. Pour cela, nous proposons de directement analyser la matrice Jacobienne $\mathbf{J}(\tilde{\mathbf{x}})$ utilisée pour l'estimation de pose (*cf.* section 2.6.3). En effet, cette matrice définit directement le mouvement des pixels de l'image par rapport aux 6 degrés de liberté de la pose. De plus elle peut être pré-calculée sur l'image de référence. La matrice $\mathbf{J}(\tilde{\mathbf{x}})$ relie directement les gradients photométriques $\mathbf{J}_{\mathcal{I}^*}$ aux gradients géométriques $\mathbf{J}_G$, (*cf.* Annexe V), tel que :

$$\mathbf{J}(\tilde{\mathbf{x}}) = \mathbf{J}_{\mathcal{I}^*}\mathbf{J}_G. \tag{5.2}$$

Elle peut être décomposée en six parties, (appellées *steepest descent images* dans Baker & Matthews (2001)) chacune correspondant à un degré de liberté du mouvement 3D :

$$\mathbf{J} = \begin{bmatrix}\mathbf{J}^1 & \mathbf{J}^2 & \mathbf{J}^3 & \mathbf{J}^4 & \mathbf{J}^5 & \mathbf{J}^6\end{bmatrix}. \tag{5.3}$$

Chaque vecteur colonne de $\mathbf{J}$ de dimensions $mn \times 1$, contient le gradient associé à une direction de mouvement, et peut être interprété comme une carte de saillance une fois ses éléments ré-arrangés sous forme matricielle. Les images de la figure 5.3, montrent les 6 images obtenues sur une sphère synthétique, un pixel clair indique un fort gradient. Les 3 premières images ($\mathbf{J}^1$,$\mathbf{J}^2$ et $\mathbf{J}^3$) correspondent aux mouvements en translation. Les points proches dans l'image ont un fort gradient. Les 3 dernières images ($\mathbf{J}^4$,$\mathbf{J}^5$ et $\mathbf{J}^6$) correspondent

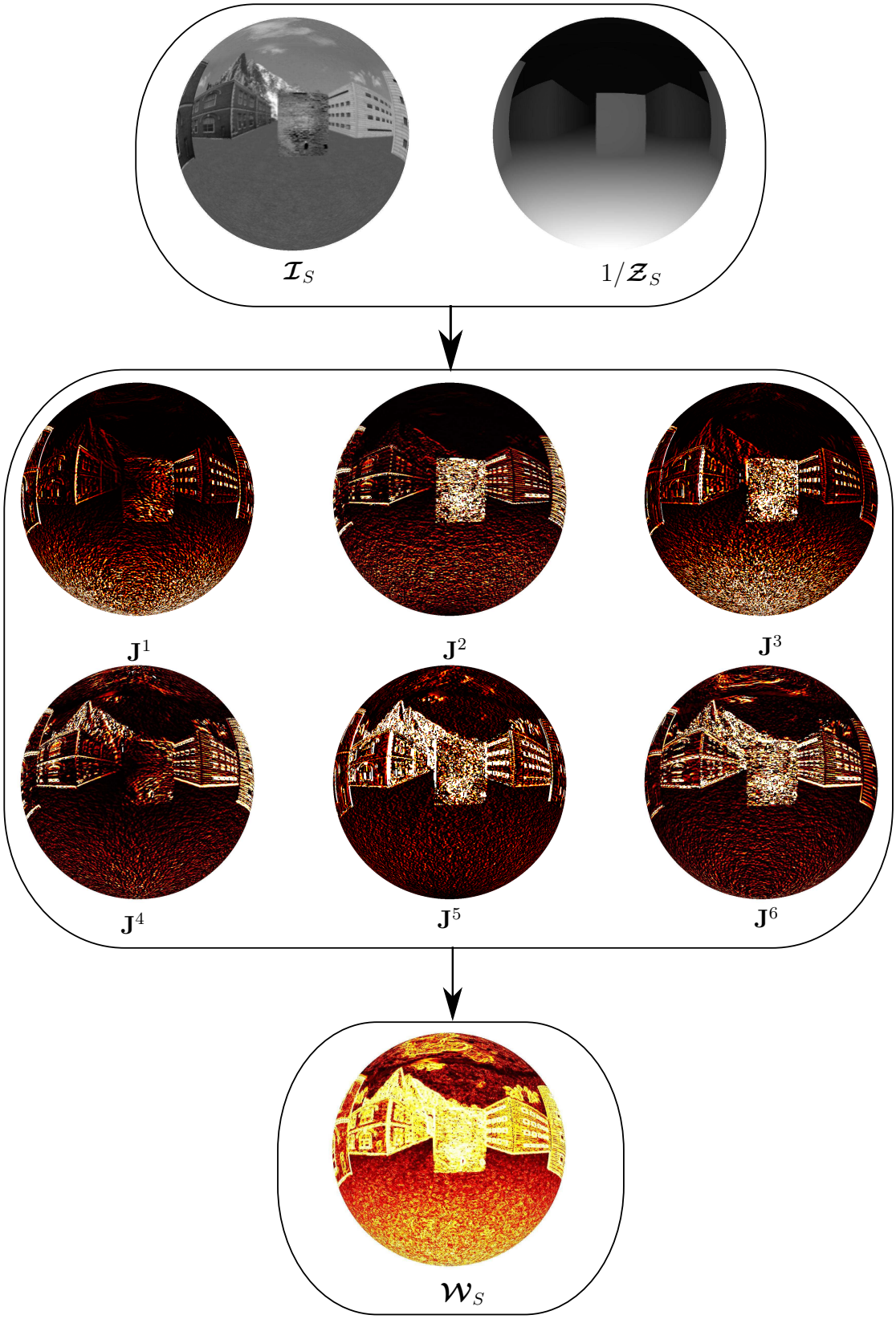


FIG. 5.3 – $\mathcal{I}_S$: Image de référence. $\mathcal{Z}_S$: Carte de profondeur. $\mathbf{J}_i$: cartes de saillance associées aux 6 degrés de liberté. $\mathcal{W}_S$: carte de saillance (une forte intensité correspond à un pixel important. À partir des 6 cartes de saillance, correspondant aux colonnes de la matrice Jacobienne, l'algorithme de sélection effectue un tri maximisant l'observabilité de chaque degrés de liberté.

aux mouvements en rotation. Dans ce cas la géométrie n'intervient pas, les points à l'infini (ciel) ont un fort gradient.

L'objectif est donc d'extraire un sous ensemble $\overline{\mathbf{J}} = \{\overline{\mathbf{J}}^1, \overline{\mathbf{J}}^2, \overline{\mathbf{J}}^3, \overline{\mathbf{J}}^4, \overline{\mathbf{J}}^5, \overline{\mathbf{J}}^6\} \subset \mathbf{J}$, de dimensions $p \times 6$, avec $p \ll nm$, contenant les pixels conditionnant le mieux chaque degré de liberté de la matrice $\mathbf{J}$.

Nous proposons un algorithme itératif de tri où chaque ligne k de la nouvelle matrice $\overline{\mathbf{J}}$ est obtenue par :

$$\overline{\mathbf{J}}_k = \arg\max_j (|\mathbf{J}_i^j| \setminus \tilde{\mathbf{J}}), \tag{5.4}$$

ce qui correspond à sélectionner une ligne entière de la matrice orignale $\mathbf{J}$, correspondant au meilleur gradient de la colonne j (le j^{ème} degrés de liberté). $\tilde{\mathbf{J}} \subset \overline{\mathbf{J}}$ est un sous ensemble intermédiaire contenant les lignes de $\mathbf{J}$ déjà sélectionnées, et $\setminus\tilde{\mathbf{J}}$ signifie qu'il n'est possible de re-sélectionner la même ligne de $\mathbf{J}$. Le terme (5.4) est répété itérativement dans chaque direction, de manière à ce qu'un nombre égal de pixels soit sélectionné pour chaque degré de liberté, jusqu'à ce que tous les pixels soient triés.

La figure 5.4 montre un exemple de sélection pour quelques valeurs d'une matrice $\mathbf{J}$ (à droite). Pour chaque colonne de $\mathbf{J}$ les pixels sont triés par ordre décroissant de 1 à mn en fonction de leur gradient. La matrice $\overline{\mathbf{J}}$ est alors construite itérativement en sélectionnant le meilleur pixel de chaque degré de liberté, si celui-ci n'a pas déjà été sélectionné.

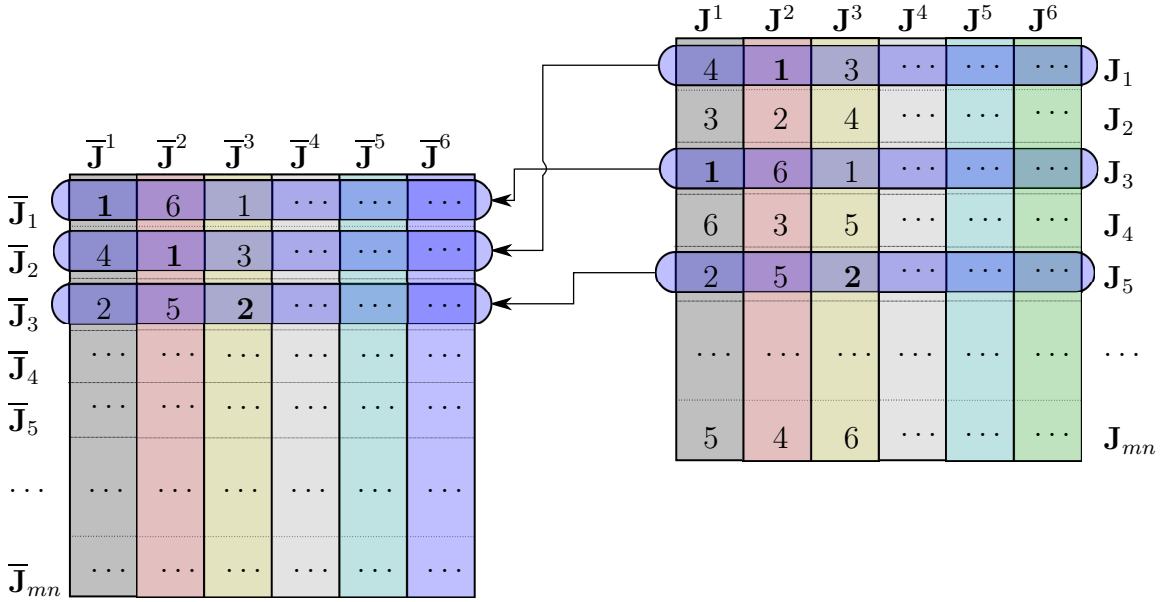


FIG. 5.4 – La matrice $\overline{\mathbf{J}}$ est construite itérativement en sélectionnant les meilleurs pixels de la matrice $\mathbf{J}$, classés pour chaque colonne de 1 à mn (du plus fort gradient au plus faible gradient).

A la fin de l'algorithme, les pixels sont triés de manière à conditionner le mieux les 6 degrés de liberté. En pratique, la matrice $\overline{\mathbf{J}}$ n'est pas explicitement construite (*cf.* algorithme 5.1), les indices des lignes de la matrice $\mathbf{J}$, correspondant aux coordonnées des pixels dans l'image, sont stockés par ordre de saillance dans la carte $\mathcal{W}_S$.

Lors de la phase d'apprentissage, la matrice Jacobienne $\mathbf{J}(\tilde{\mathbf{x}})$ est calculée sur l'image de référence. Tous les pixels de l'image sont alors triés suivant le critère (5.4) et les indices des meilleurs pixels sont stockés dans le vecteur $\mathcal{W}_S$ et enregistrés dans la base données. Lors

Algorithme 5.1 Algorithme de sélection

Entrées: $\mathbf{J} \in \mathbb{R}^{n\times 6}$
Sorties: $\boldsymbol{\mathcal{W}}_S \in \mathbb{N}^n$
  $k \leftarrow 1$
  pour $i = 1 \rightarrow n/6$ **faire**
    pour $j = 1 \rightarrow 6$ **faire**
      $indice \leftarrow \arg\max_i(\mathbf{J}_i^j \setminus \tilde{\mathbf{J}})$
      pour $j = 1 \rightarrow 6$ **faire**
        $\tilde{\mathbf{J}}_k^j \leftarrow \mathbf{J}_{indice}^j$
      fin pour
      $\boldsymbol{\mathcal{W}}_S(k) \leftarrow indice$
      $k \leftarrow k + 1$
    fin pour
  fin pour
  Retourner $\boldsymbol{\mathcal{W}}_S$.

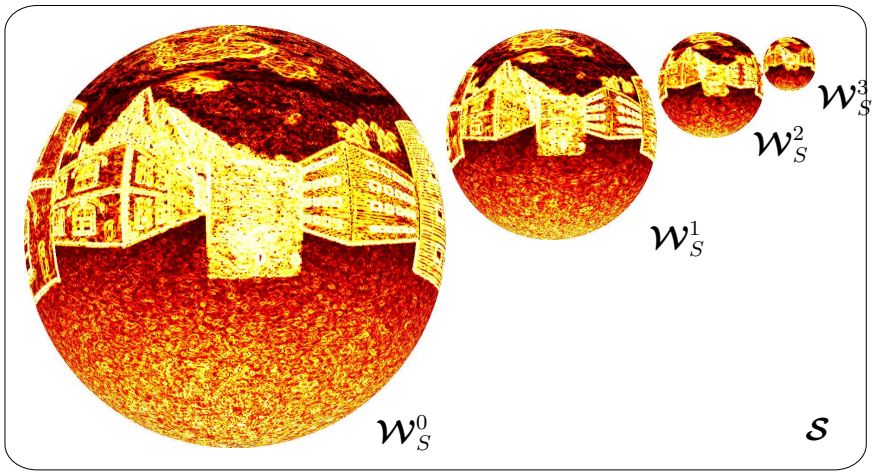


FIG. 5.5 – Sélection des pixels effectuée sur chaque résolution de la sphère.

de la phase de localisation en ligne (*cf.* section 6.2), une sélection dynamique des pixels est effectuée en fonction de la carte de saillance pré-calculée, et du point de vue de la caméra courante, ce qui permet d'utiliser seulement les p premiers points désirés tout en conservant l'observabilité des 6 degrés de liberté. Cette sélection sera modélisée par la suite dans les fonctions de *warping*, par la fonction de saillance $s(.)$:

$$(Z^s, \mathbf{q}^s) = s(Z, \mathbf{q}), \tag{5.5}$$

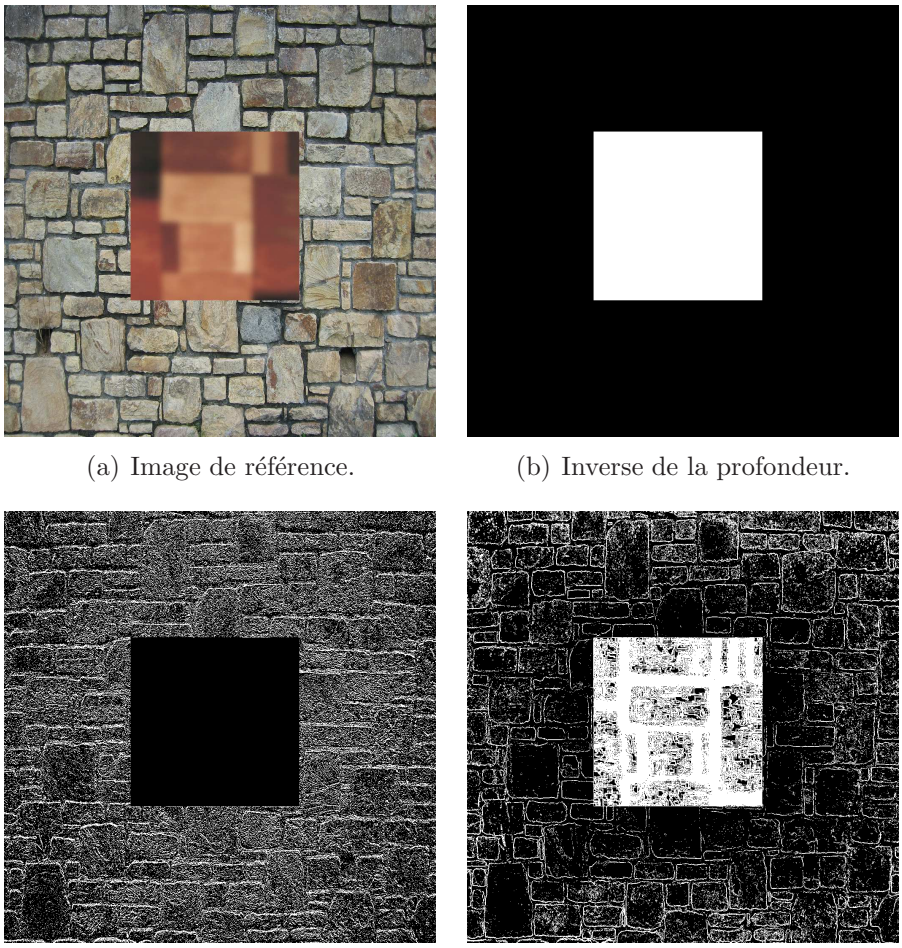
permettant d'utiliser le meilleur pixel de la carte de tri $\boldsymbol{\mathcal{W}}_S$, de manière efficace en utilisant une *LUT* (*Look up table*).

Puisqu'une approche multi-résolution est utilisée lors de l'alignement d'images (*cf.* section 2.7), l'algorithme de sélection est appliqué sur chaque résolution de la sphère (*cf.* figure 5.5).

5.3.2 Résultats de simulations

5.3.2.1 Plan à l'infini

Afin d'illustrer l'importance de la géométrie dans la sélection d'information, une simulation sur image synthétique a été effectuée. L'image de référence $\mathcal{I}^*$ contient un plan proche peu texturé, et un plan à l'infini contenant de fort gradients photométriques (cf. Fig. 5.6). L'image courante utilisée est la même image, mais l'algorithme d'estimation de pose est initialisé avec une erreur en translation et une erreur en rotation. Trois approches ont étés comparées : un recalage dense, c'est à dire en utilisant tous les pixels disponibles, un recalage utilisant 25 % des meilleurs gradients photométriques (GP), affichés en blanc sur la figure 5.6(c), et un recalage utilisant 25 % des meilleurs pixels obtenus par la méthode proposée utilisant à la fois les gradients photométriques et les gradients géométriques (GP+GG) (cf. Fig. 5.6(d)).

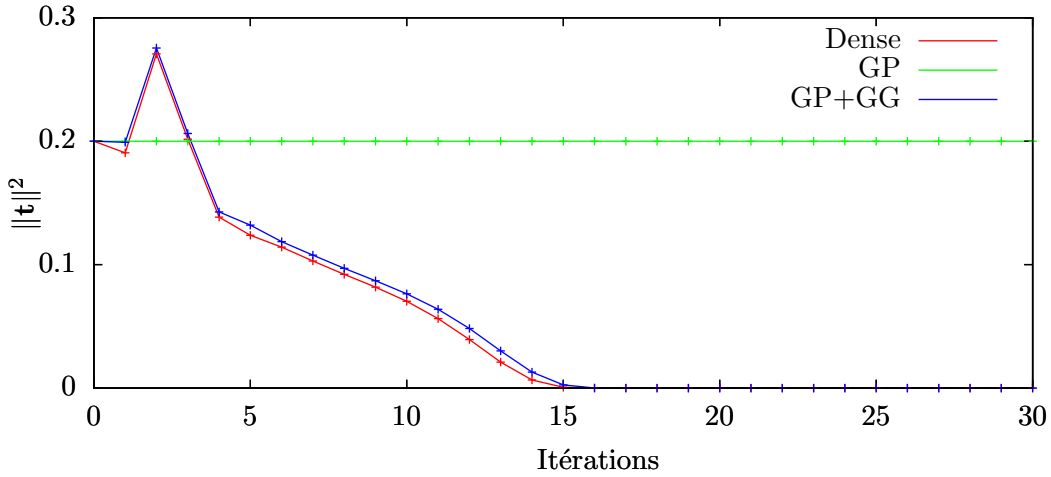


(a) Image de référence. (b) Inverse de la profondeur.

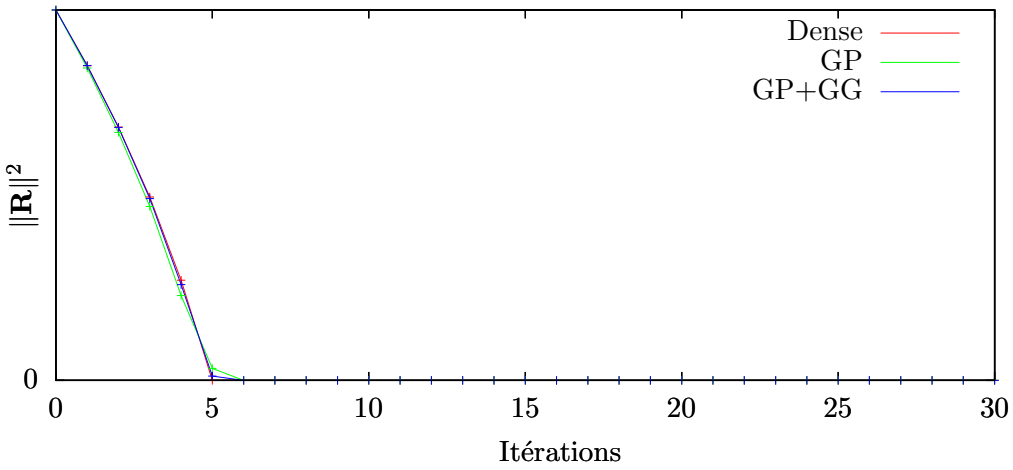
(c) Gradients photométriques (GP). (d) Méthode proposée (GG+GP).

Fig. 5.6 – Simulation d'un plan texturé à l'infini et d'un plan peu texturé proche de la caméra. Sélection de 25 % des meilleurs pixels. La méthode basée uniquement sur les gradients photométriques (c), sélectionne seulement les points à l'infini (affichés en blanc). La méthode proposée (d), combine les gradients photométriques et les gradients géométriques, et permet de sélectionner aussi les pixels du premier plan (affichés en blanc).

La figure 5.7 montre la décroissance de la norme de l'erreur de la pose estimée en fonction du nombre d'itérations, pour les translations : $\|\mathbf{t}\|^2$ et les rotations : $\|\mathbf{R}\|^2$. Pour le mouvement en rotation, les trois méthodes convergent logiquement vers la solution car l'estimation de la rotation est indépendante de la profondeur. Concernant l'estimation de la translation, la méthode dense et la méthode proposée convergent vers le minimum global. Comme prévu, la méthode basée sur la sélection photométrique n'a pas sélectionné de points observables en translation, elle est donc impossible à estimer.



(a) Norme de l'erreur en translation.



(b) Norme de l'erreur en rotation.

Fig. 5.7 – Comparaison des méthodes : dense, gradients photométriques (GP) et gradients photométriques + gradients géométriques (GP+GG). Les trois méthodes permettent d'estimer le mouvement en rotation (b). Pour la translation (a), la méthode GP n'a pas sélectionné de pixels observables. La méthode proposée permet d'estimer les 6 degrés de liberté, avec une vitesse de convergence très proche de la méthode dense (utilisant tous les pixels de l'image de référence).

5.3.2.2 Temps de calculs

En terme de temps de calcul, la complexité de l'algorithme de minimisation (*cf.* section 2.6) est en $\mathcal{O}(p)$, le gain est directement proportionnel au nombre de pixels p utilisés dans la minimisation. Les résultats de la figure 5.8 montrent le temps de calcul d'une itération de l'algorithme de recalage en fonction du pourcentage de pixels sélectionnés ($n_{max} = 8.10^4$ pixels). Alors qu'il faut 3.6 ms à l'algorithme pour effectuer une itération en utilisant tous les pixels de l'image de référence, le temps de calcul est réduit à 0.9 ms en utilisant 25% des meilleurs pixels.

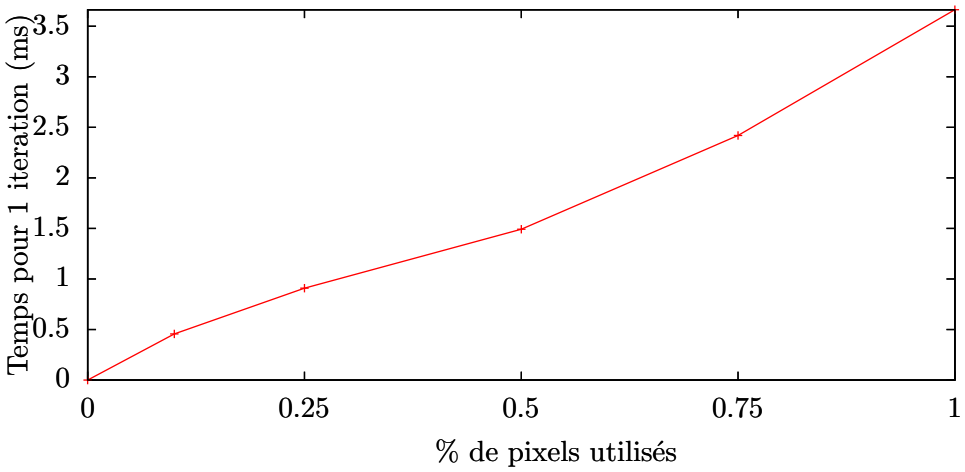


FIG. 5.8 – Temps de calcul avec sélection d'information. Le temps de calcul est proportionnel aux nombres de pixels utilisés dans la minimisation.

5.4 Conclusion

La nouvelle méthode de sélection de pixels proposée ici est adaptée aux techniques directes. Contrairement aux méthodes classiques d'extraction de points d'intérêts, la technique développée se base sur l'analyse de la matrice Jacobienne utilisée pour l'estimation itérative du déplacement. Cette matrice combine à la fois les gradients photométriques et les gradients géométriques associés à chaque pixel. Par rapport aux approches précédentes basées sur une information locale 2D, ce type de sélection permet de maximiser l'observabilité des déplacements pour les 6 degrés de liberté de la pose et de conserver un maximum de précision. De plus la sélection est directement effectuée hors-ligne, lors de la construction de la base de donnée. Lors de la phase de localisation en ligne, la carte de saillance est utilisée de manière efficace pour sélectionner un nombre réduit de pixels dans la fonction de recalage d'image, ce qui permet d'effectuer les calculs en temps-réel, à haute fréquence (*i.e.* 45 Hz).

L'algorithme proposé est très simple à mettre en oeuvre et efficace à calculer, cependant il serait intéressant d'analyser les effets d'un changement de base de la matrice $\mathbf{J}$, tel qu'une *SVD* (*Singular value decomposition*) ou une *PCA* (*Principal component analysis*), qui permettent de projeter la matrice sur une base orthogonale, et donc de décorréler ses éléments. Quelques expérimentations ont été menées sur ces transformées lors du développement de l'algorithme proposé (*i.e.* sélection des pixels sur la *SVD* de $\mathbf{J}$), cependant les

résultats constatés sont très peu de différents de ceux obtenus par la sélection effectuée directement sur la matrice **J**. Néanmoins il serait intéressant d'effectuer une étude théorique plus approfondie sur les avantages de ces décompositions.

Troisième partie

Localisation en ligne et navigation autonome

Chapitre 6

Localisation temps réel

6.1 Introduction

Ce chapitre aborde l'utilisation en ligne du graphe d'images augmentées, c'est à dire la localisation en temps réel d'une caméra, montée sur un robot naviguant dans le voisinage du graphe d'apprentissage. La localisation de la caméra est définie comme un problème de minimisation entre les intensités de l'image courante et celles de l'image de référence la plus proche (nœud du graphe) sélectionnée suivant un critère prenant en compte la largeur du champ visuel. Pour une meilleure robustesse aux mouvements en rotation, une méthode d'estimation efficace du changement d'orientation 3D de la caméra entre deux images successives est utilisée. Enfin pour une obtenir une meilleure robustesse et une meilleure précision, il est proposé d'utiliser simultanément plusieurs nœuds du graphe dans la localisation.

6.2 Localisation

Soit l'image $\boldsymbol{\mathcal{I}}_t$, acquise par une caméra à l'instant t (e.g monoculaire, stéréo ou omni-directionnelle) et une pose initiale $\widehat{\mathbf{T}}_{\mathcal{G}}$ (*i.e.* dernière pose connue), exprimée dans le repère du graphe $\mathcal{G}$ de la base de données. Cette pose initiale permet l'extraction de l'image de référence augmentée $\boldsymbol{\mathcal{S}}^* = \{\boldsymbol{\mathcal{I}}^*, \boldsymbol{\mathcal{Z}}^*, \boldsymbol{\mathcal{P}}^*, \boldsymbol{\mathcal{W}}^*\}$ la plus "proche" et de sa pose $\mathbf{T}^*_{\mathcal{G}}$. Des détails supplémentaires sur le choix de l'image de référence sont donnés en section 6.3.

Dans le repère de l'image de référence $\mathbf{S}^*$ sélectionnée, la pose initiale $\widehat{\mathbf{T}}_{\mathbf{S}}$ s'écrit :

$$\widehat{\mathbf{T}}_{\mathbf{S}} = (\mathbf{T}^*_{\mathcal{G}})^{-1}\widehat{\mathbf{T}}_{\mathcal{G}}. \tag{6.1}$$

L'objectif est d'estimer la transformation incrémentale $\mathbf{T}(\mathbf{x})$, minimisant les erreurs d'intensités entre l'image courante et l'image de référence, tel qu'illustré sur la figure 6.1 :

$$\mathbf{e}(\mathbf{x}) = \boldsymbol{\mathcal{I}}\Big(w(\widehat{\mathbf{T}}_{\mathbf{S}}\mathbf{T}(\mathbf{x}); s(Z, \mathbf{p}^*))\Big) - \boldsymbol{\mathcal{I}}^*\Big(s(Z, \mathbf{p}^*)\Big), \tag{6.2}$$

où la fonction $s(Z, \mathbf{p}^*)$ permet de sélectionner les meilleurs pixels de l'image $\boldsymbol{\mathcal{I}}^*$ en fonction de la carte de saillance $\boldsymbol{\mathcal{W}}^*$ pré-calculée sur la sphère de référence $\boldsymbol{\mathcal{S}}^*$ selon l'algorithme de sélection de la section 5.3. Cette erreur est minimisée par la méthode robuste d'alignement d'image directe définie en section 2.6, afin d'obtenir la pose de la caméra $\mathbf{T}_{\mathbf{S}}$ et donc du véhicule. Il est alors possible de mettre à jour la pose $\widehat{\mathbf{T}}_{\mathcal{G}}$, exprimée dans le repère du graphe.

Grâce à la sélection d'information, effectuée lors de l'apprentissage hors-ligne, seulement un nombre p des pixels sont utilisés dans la fonction d'erreur, permettant d'accélérer considérablement le temps de calcul, sans dégrader la précision, ni l'observabilité.

6.3 Sélection de l'image de référence

6.3.1 Cas d'un graphe d'images sphériques

Puisqu'une image sphérique fournit toute l'information nécessaire à la localisation autour d'un point de vue, la sphère de référence la plus proche choisie pour la localisation est naturellement sélectionnée par la norme de la distance Euclidienne en translation, entre la pose initiale $\widehat{\mathbf{T}}$ et les poses du graphe :

$$\mathbf{T}_{\mathbf{S}}^{*} = \arg\min_{\mathbf{i}}(\|(\mathbf{T}_{\mathcal{G}}^{i})^{-1}\widehat{\mathbf{T}}_{\mathcal{G}}\mathbf{e}_4^T\|) \quad \forall i \in \mathcal{G}, \tag{6.3}$$

où le vecteur de coordonnées homogènes $\mathbf{e}_4 = \begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T$ permet d'extraire le vecteur de translation de la i^{ieme} pose du graphe $\mathbf{T}_{\mathbf{S}}^{i} = (\mathbf{T}_{\mathcal{G}}^{i})^{-1}\widehat{\mathbf{T}}_{\mathcal{G}}$.

Sélectionner la sphère visuelle la plus proche, permet d'une part de minimiser la quantité d'occultation des bâtiments, proportionnelle à la distance entre les points de vue, et d'autre part les différences de résolution entre l'image de référence et l'image courante, ce qui améliore les performances des techniques d'alignement d'image directes.

6.3.2 Cas d'un graphe d'images perspectives

Dans certaines séquences d'images traitées dans cette thèse, seulement des images stéréo classiques étaient disponibles pour construire le graphe d'apprentissage. Avec ce type de capteur, l'angle de vue des images est limité et la sélection de l'image de référence est plus délicate que dans le cas d'une sphère. En effet il faut en permanence assurer un recouvrement entre l'image de référence et l'image vue par la caméra courante. Par exemple deux images perspective capturées dans deux directions opposées mais à la même position, ont une distance en translation nulle, mais n'ont aucun recouvrement. Il est donc indispensable de prendre en compte l'orientation. Cependant, maximiser le recouvrement n'est pas non plus suffisant car il est possible de sélectionner une image de référence très distante (*e.g.* loin devant), ce qui génère des problèmes d'occultation et des différences en résolution trop importantes, ce qui peut empêcher un alignement correct des images. Une solution optimale serait d'effectuer un *clipping*, c'est à dire projeter et tester l'appartenance de tous les points 3D de chaque image de référence à la position courante. Cependant cette approche est bien trop coûteuse pour être utilisée en temps réel. La solution retenue permet de prendre en compte la translation et l'orientation de l'image de référence, en minimisant la distance vectorielle mesurée entre deux vecteurs formés par les axes optiques des deux images :

$$\mathbf{T}_{\mathcal{G}}^{*} = \arg\min_{i}(\|(\mathbf{T}_{\mathcal{G}}^{i})^{-1}\widehat{\mathbf{T}}_{\mathcal{G}}\overline{\mathbf{P}} - \overline{\mathbf{P}}\|) \quad \forall i \in \mathcal{G} \tag{6.4}$$

où le point en coordonnées homogènes $\overline{\mathbf{P}} = \begin{bmatrix}0 & 0 & d & 1\end{bmatrix}^T \in \mathbb{R}^4$ est un point 3D positionné sur l'axe optique de l'image de référence, à une distance d. Plus la distance d est importante, plus l'orientation est privilégiée, au détriment de la distance. En revanche, plus d est faible, plus la translation est privilégiée, au détriment du recouvrement. Dans le cas où $d = 0$, la sélection est identique à la sélection d'un graphe sphérique. Lors des expérimentations, une

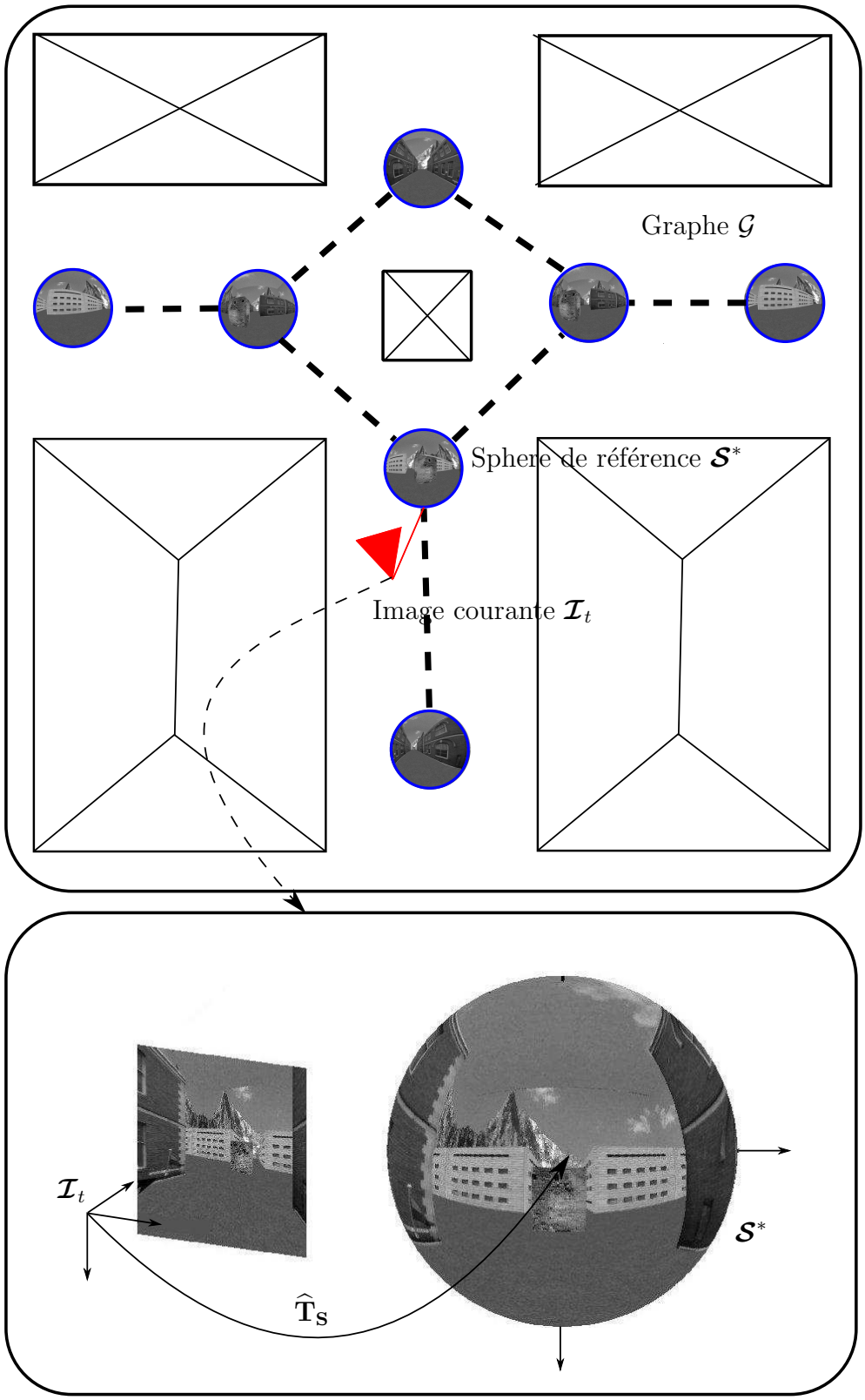


FIG. 6.1 – Localisation en ligne : l'image courante $\mathcal{I}_t$ est recalée sur la sphère de référence la plus proche.

distance d de 5 mètres a été utilisée. Cette solution n'est certainement pas optimale, mais permet une sélection très rapide de l'image de référence, et s'est montrée très efficace lors des expérimentations que ce soit en environnement urbain le long d'une trajectoire, ou en intérieur localement dans une pièce.

6.4 Estimation efficace de la rotation 3D locale

Typiquement, le mouvement apparent des pixels dans une image est dominé par le mouvement en rotation de la caméra. En effet, pour une rotation pure, le déplacement des pixels de l'image est indépendant de la géométrie de la scène. Pour un algorithme de localisation visuelle (direct ou basé point d'intérêts), de larges mouvements en rotation entre deux images successives sont très souvent la cause d'échecs. Afin de mieux initialiser la localisation 3D, il est possible d'effectuer au préalable une estimation du changement d'orientation 3D de la caméra avec une méthode directe tel que dans Mei *et al.* (2010); Lovegrove & Davison (2010); Newcombe *et al.* (2011b).

Pour cela, la plus petite image de la pyramide multi-résolution est utilisée (cf. 2.7). Cette image, issue de filtrages Gaussiens et de sous échantillonnages successifs, peut être associée à une caméra invariante aux mouvements en translation, car la distance focale f de la matrice des paramètres intrinsèques $\mathbf{K}$ est très faible (divisée successivement par un facteur 2). En d'autres termes, la géométrie de la scène peut être considérée comme étant à l'infini et donc les translations locales négligées. Il est alors possible d'effectuer un alignement direct, en rotation pure, entre l'image acquise à l'instant t : $\boldsymbol{\mathcal{I}}_t$, et l'image acquise à l'instant $t-1$: $\boldsymbol{\mathcal{I}}_{t-1}$, en minimisant les erreurs d'intensités suivantes :

$$\mathbf{e}(\mathbf{x}_\omega) = \boldsymbol{\mathcal{I}}_t\left(w(\mathbf{K}\widehat{\mathbf{R}}\mathbf{R}(\mathbf{x}_\omega)\mathbf{K}^{-1};\mathbf{p})\right) - \boldsymbol{\mathcal{I}}_{t-1}(\mathbf{p}). \tag{6.5}$$

où le vecteur $\mathbf{x}_\omega \in \mathbb{R}^3$ contient seulement les vitesses angulaires, et la matrice $\mathbf{R}(\mathbf{x}_\omega)$ est obtenue à partir de l'application matrice exponentielle de la matrice antisymétrique de $\mathbf{x}_\omega$ (*cf.* section 2.2.2).

Le matrice $\mathbf{H} = \mathbf{K}\widehat{\mathbf{R}}\mathbf{R}(\mathbf{x}_\omega)\mathbf{K}^{-1} \in \mathbb{SL}(3)$ est une matrice d'homographie (*cf.* Hartley & Zisserman (2004)), permettant de transférer les pixels $\mathbf{p}$ de l'image $\boldsymbol{\mathcal{I}}_t$ vers l'image $\boldsymbol{\mathcal{I}}_{t-1}$, sans utiliser l'information de profondeur ($Z \to \infty$).

Cette fonction d'erreur peut être minimisée par une méthode des moindres carrées, de type inverse compositionnelle ou *ESM*, afin de mettre à jour incrémentalement la valeur de la rotation $\widehat{\mathbf{R}}$ par :

$$\widehat{\mathbf{R}} \leftarrow \widehat{\mathbf{R}}\mathbf{R}(\mathbf{x}_\omega) \tag{6.6}$$

Au final, cette minimisation est quasiment identique à une estimation complète des 6 degrés de liberté 2.6, mais seulement les 3 degrés de rotation sont estimés. De plus avec un modèle géométrique parfait (points à l'infini), la convergence vers la solution est très rapide. En pratique, quelques itérations (≈ 5) sont suffisantes pour estimer des mouvements rotationnels forts (*i.e.* caméra excitée manuellement). La matrice $\widehat{\mathbf{R}}$ obtenue, est ensuite utilisée pour mettre à jour l'initialisation de la pose 3D de l'image courante par :

$$\widehat{T}_{\mathcal{G}} \leftarrow \widehat{T}_{\mathcal{G}} \begin{bmatrix} \widehat{\mathbf{R}} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{6.7}$$

ce qui permet d'initialiser le suivi à 6 degrés de liberté entre l'image de référence de la base de données et l'image courante plus proche de la solution.

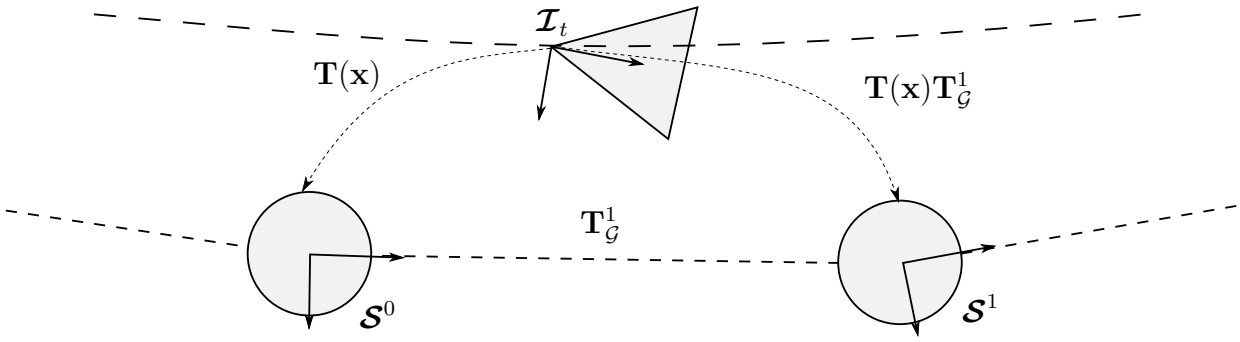


FIG. 6.2 – Schéma d'utilisation simultanée de plusieurs nœuds du graphe. La localisation de l'image $\mathcal{I}_t$ est effectuée avec les nœuds $\mathcal{S}^0$ et $\mathcal{S}^1$.

6.5 Utilisation de plusieurs nœuds du graphe

Lors de la phase de localisation, l'image courante est en permanence recalée (*cf.* équation (6.2)) sur l'image de référence la plus proche de la base de donnée, extraite suivant le critère (6.3). La même image de référence est alors conservée pendant un certain temps jusqu'à ce qu'une nouvelle image soit sélectionnée en fonction de la pose courante et du critère de sélection.

Cependant, plus la distance absolue entre l'image de référence et l'image courante augmente, moins l'hypothèse Lambertienne des points de la scène est vérifiée, et plus les occultations sont présentes, ce qui génère des erreurs dans la fonction de coût, rendant l'estimation de pose moins précise. Ces erreurs, ajoutées à l'incertitude sur le positionnement des images de référence, peuvent générer des discontinuités sur les trajectoires estimées lors du changement d'image de référence (*cf.* figure 6.3). Quoique souvent minimes (de l'ordre de quelques dizaines de centimètres), ces décrochages peuvent être gênants pour effectuer l'asservissement visuel d'un véhicule, et peuvent même dans certains cas conduire à initialiser l'alignement de l'image suivante trop loin de la solution, ce qui peut empêcher l'algorithme de converger.

Afin d'obtenir des trajectoires les plus lisses possibles, il est souhaitable d'utiliser deux images de référence simultanément dans la minimisation, sélectionnées successivement selon le critère de l'équation (6.3), ce qui correspond à sélection l'image précédente et l'image suivante du graphe comme illustré sur la figure 6.2. Dans ce cas, le problème de localisation consiste à minimiser simultanément les deux erreurs d'intensités suivantes :

$$\mathbf{e}_{\mathbf{G}}(\mathbf{x}) = \begin{bmatrix} \mathcal{I}_t\Big(w(\widehat{\mathbf{T}}_{\mathbf{S}}\mathbf{T}(\mathbf{x})\mathbf{I}; s(Z^0, \mathbf{p}^0))\Big) - \mathcal{I}^0\Big(s(Z^0, \mathbf{p}^0)\Big) \\ \mathcal{I}_t\Big(w(\widehat{\mathbf{T}}_{\mathbf{S}}\mathbf{T}(\mathbf{x})\mathbf{T}^1_{\mathcal{G}}; s(Z^1, \mathbf{p}^1))\Big) - \mathcal{I}^1\Big(s(Z^1, \mathbf{p}^1)\Big) \end{bmatrix}, \tag{6.8}$$

où la pose $\mathbf{T}^1_{\mathcal{G}}$ définit la pose de la seconde image de référence $\mathbf{S}^1$ exprimée dans le repère de la première image de référence $\mathbf{S}^0$ (*cf.* figure 6.2).

Cette minimisation consiste à empiler les vecteurs d'erreur, dans un seul vecteur global $\mathbf{e}_{\mathbf{G}}(\mathbf{x})$, et les matrices Jacobiennes dans une matrice globale par :

$$\mathbf{J}_G = \begin{bmatrix} \mathbf{J}_0 \\ \mathbf{J}_1\mathbf{J}_v(\mathbf{T}^1_{\mathcal{G}}) \end{bmatrix}, \tag{6.9}$$

où $\mathbf{J}_v(\mathbf{T}^1_{\mathcal{G}})$ est la matrice adjointe associée à la pose $\mathbf{T}^1_{\mathcal{G}}$, permettant de transformer les vitesses estimées du repère de $\mathbf{S}^1$ vers le repère de $\mathbf{S}^0$ tel que :

$$\mathbf{J}_v(\mathbf{T}) = \begin{bmatrix} \mathbf{R} & \mathbf{t}_\times \mathbf{R} \\ \mathbf{0}_3 & \mathbf{R} \end{bmatrix}, \tag{6.10}$$

l'opérateur $[.]_\times$ étant l'opérateur de la matrice antisymétrique (cf. section 2.2.2).

Utiliser deux images de référence simultanément, signifie utiliser deux fois plus de pixels dans la fonction de coût. Pour conserver la même fréquence de calcul, le nombre de pixels utilisés par la sélection d'information est simplement réparti également entre les deux images.

L'équation (6.8) est facilement extensible à n images de référence. Cependant, lors des expérimentations, l'utilisation d'un nombre d'images $n > 2$, n'a pas montré de résultats convaincants. En effet l'utilisation d'images de référence trop distantes de l'image courante introduit des occultations, ce qui réduit l'efficacité de l'algorithme.

6.5.1 Comparaison mono/multi modèle

La figure 6.3 montre une portion de l'estimation de la trajectoire d'une caméra monoculaire, montée sur un véhicule naviguant localement dans un graphe d'images augmentées. A l'initialisation la position du véhicule est $(X = 0, Z = 0)$. Puis, le véhicule se déplace pour aller vers la position $(X = -10, Z = 13)$. Deux estimations ont étés effectuées :

- La première (rouge) utilise seulement un nœud du graphe dans la localisation, sélectionné suivant le critère (6.3). Lors du changement d'image de référence, au point $(X = -0.9, Z = 5)$, on peut voir que la trajectoire n'est pas continue, le mouvement estimé entre deux images successives correspond à un mouvement latéral du véhicule, ce qui n'est pas possible.
- La seconde (bleu) utilise simultanément 2 nœuds du graphe dans la localisation (*cf.* équation (6.8)). Lors du changement d'image de référence, la trajectoire estimée est continue sur l'ensemble de la séquence, et reste cohérente avec le mouvement global du véhicule.

6.6 Résultats

6.6.1 Implémentation

Une implémentation temps-réel de l'algorithme de localisation en ligne a été réalisée en C++ et permet de localiser une caméra de résolution 800×600 pixels à une fréquence de 45 Hz. Le tableau 6.1 montre les étapes successives de l'algorithme, ainsi que les temps de calculs obtenus sur un ordinateur portable équipé d'un processeur Intel core i7-2820QM.

Étape	Temps (ms)
Recherche image référence	<0.01 ms
Construction pyramide gaussienne	0.70 ms
Estimation rotation inter-images	0.48 ms / itération
Estimation 6 degrés de liberté	1.13 ms / itération

Tab. 6.1 – Étapes et temps de calcul de l'algorithme de localisation temps réel pour $6e10^4$ pixels.

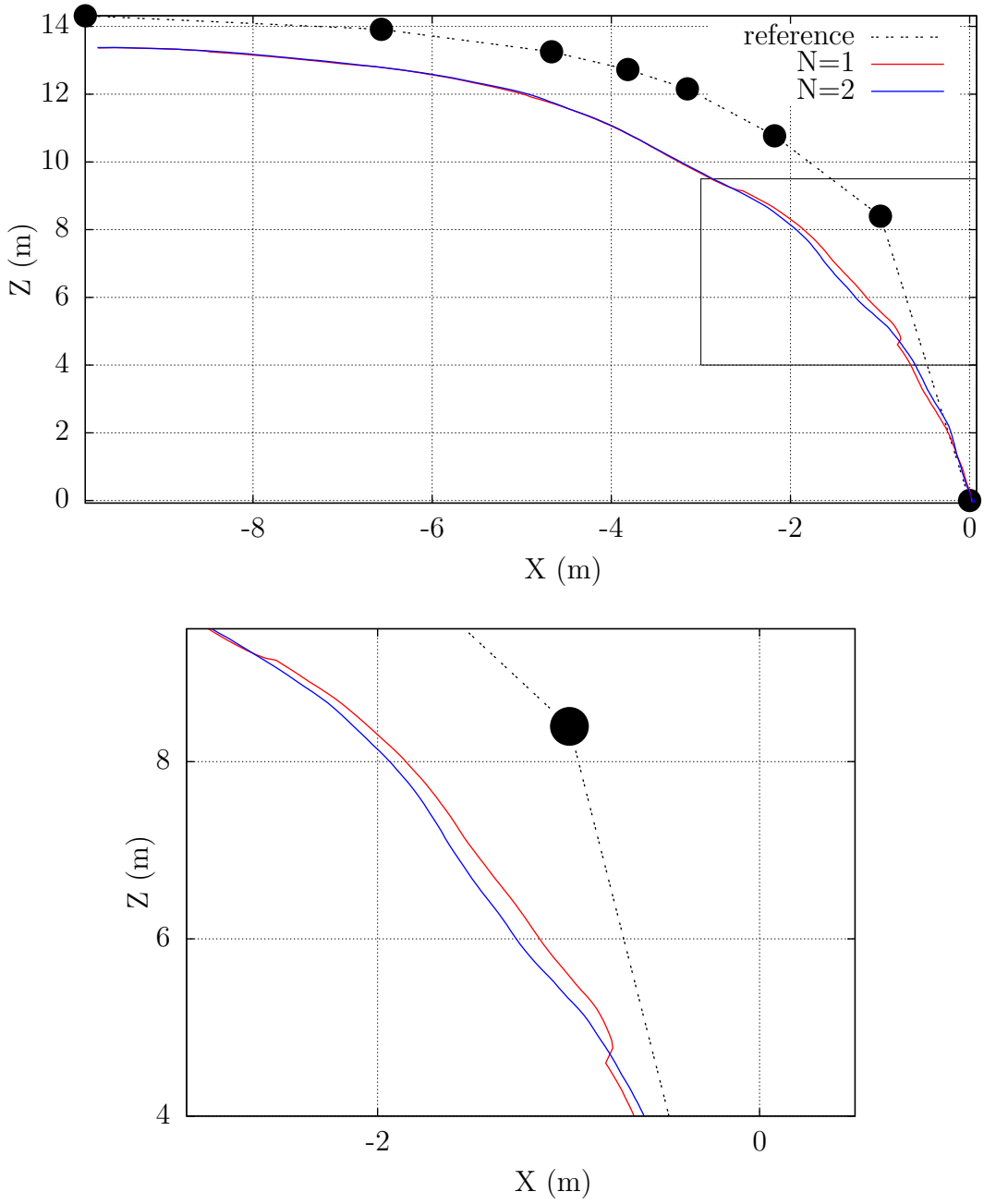


FIG. 6.3 – Utilisation simultanée de plusieurs nœuds du graphe : Les disques noirs indiquent la position des images de référence. En rouge, la trajectoire estimée en utilisant une seule image de référence. La trajectoire contient des décrochages lors du changement d'image de référence. En bleu, la même trajectoire estimée avec 2 images de référence. La trajectoire obtenue est continue, sans décrochage lors des changements d'image de référence.

Une attention particulière à été porté sur la parallélisation des calculs, afin de profiter des processeurs multi-cœurs, aujourd'hui présents sur tous les ordinateurs modernes. La méthode de minimisation est en effet fortement parallélisable (*warping*, erreur, pseudo-inverse). Le calcul de la médiane, utilisée dans la fonction de coût robuste (*cf.* équation (2.45)), nécessite en théorie le tri des $p/2$ premiers pixels du vecteur d'erreur à chaque itération de la minimisation, ce qui peut être couteux. Cependant, puisque l'erreur est bornée (*i.e.* $[-255; 255]$), il est possible d'obtenir une bonne approximation de la valeur médiane de l'erreur en utilisant des histogrammes, ce qui réduit considérablement le coût calculatoire. Le calcul robuste de la matrice Hessienne $\mathbf{J}^T\mathbf{D}\mathbf{J}$ et du gradient $\mathbf{J}^T\mathbf{D}\mathbf{e}(\mathbf{x})$ peuvent également être parallélisés par :

$$\mathbf{J}^T\mathbf{D}\mathbf{J} = \sum_{i=0}^{n} \mathbf{J}_i^T\mathbf{D}_i\mathbf{J}_i, \tag{6.11}$$

et

$$\mathbf{J}^T\mathbf{D}\mathbf{e}(\mathbf{x}) = \sum_{i=0}^{n} \mathbf{J}_i^T\mathbf{D}_i\mathbf{e_i}(\mathbf{x}). \tag{6.12}$$

où n représente le nombre de "blocs" de pixels traités en parallèle.

Une deuxième optimisation particulièrement efficace a été l'utilisation d'instructions SIMD (*Single Instruction Multiple Data*), permettant par exemple d'effectuer simultanément 4 opérations en réels flottants pour une seule instruction processeur, ce qui permet d'accélérer d'un facteur 4 certains calculs trop complexes pour le compilateur.

L'algorithme de localisation se déroule de la manière suivante : pour chaque image courante, les images de référence augmentées les plus proches (pré-chargées en RAM) sont sélectionnées suivant le critère (6.3) en fonction de la pose initiale. Une pyramide Gaussienne est ensuite construite (*cf.* section 2.7) pour l'image courante $\boldsymbol{\mathcal{I}}_t$. Le mouvement local en rotation de la caméra est alors estimé en minimisant la fonction d'erreur (6.5) sur la plus petite échelle de la pyramide. Puisque ce suivi implique des images de très faible résolution (*i.e.* 100×75), tous les pixels peuvent être utilisés dans la minimisation. Finalement une estimation précise des 6 degrés de liberté de la pose est effectuée en utilisant un faible nombre de pixels saillants (environ $6e10^4$). La pose obtenue dans le repère du graphe est alors utilisée pour initialiser l'alignement de l'image suivante.

Le fait d'acquérir les images courantes à une fréquence de 45 Hz permet de réduire les mouvements entre deux images successives, et ainsi le nombre d'itérations nécessaire à la convergence de l'algorithme de recalage d'images. La méthode de localisation fonctionne sans modèle prédictif du mouvement de la caméra, ce qui permet d'être robuste aux mouvements non continus d'un véhicule, tels que les bosses ou les trous sur la chaussée. De plus le même algorithme peut être employé pour la localisation d'une caméra montée sur un véhicule naviguant à l'extérieur, ou d'une caméra déplacée manuellement dans un environnement intérieur.

6.6.2 Initialisation

Une méthode d'initialisation et de re-localisation a été implémentée, afin de démarrer l'algorithme de localisation, où de relancer une initialisation en cas d'échec. A l'instant $t = 0$, la caméra est considérée dans le voisinage du graphe. Un alignement dense est alors effectué sur toutes les images du graphe d'apprentissage, en utilisant la plus petite résolution de la pyramide multi-résolution et en initialisant le recalage à l'identité. Après alignement

de chaque image du graphe avec l'image courante, le score de corrélation croisée normalisée (NCC) est calculé entre l'image de référence $\boldsymbol{\mathcal{I}}^*$ et l'image recalée :

$$R = \frac{\sum_{p^*} \left(\boldsymbol{\mathcal{I}}(w(\widehat{\mathbf{T}}, Z, \mathbf{p}^*)) - \overline{\mu}\right)\left(\boldsymbol{\mathcal{I}}^*(\mathbf{p}^*) - \overline{\mu}^*\right)}{\sqrt{\sum_{p^*} \left(\boldsymbol{\mathcal{I}}(w(\widehat{\mathbf{T}}, Z, \mathbf{p}^*)) - \overline{\mu}\right)^2}\sqrt{\sum_{p^*} \left(\boldsymbol{\mathcal{I}}^*(\mathbf{p}^*) - \overline{\mu}^*\right)^2}} \tag{6.13}$$

où $\overline{\mu}$ et $\overline{\mu}^*$ sont respectivement les moyennes de l'image recalée et de l'image de référence, $\widehat{\mathbf{T}}$ est la pose après alignement et $R \in [-1; 1]$. L'image de référence choisie pour la localisation est celle avec le meilleur score de corrélation. Cette méthode d'initialisation n'est pas bien adaptée à de grandes bases de données, cependant pour des images courantes de dimensions 800×600 l'algorithme prend moins de 30 ms par image de référence, soit 9 secondes d'initialisation pour un graphe d'apprentissage de 300 images. Il serait néanmoins intéressant d'utiliser une technique dédiée à la reconnaissance de lieux telle que Cummins & Newman (2008); Chapoulie *et al.* (2011).

6.6.3 Environnements urbains

6.6.3.1 Localisation sur une sphère

Dans un premier temps, l'algorithme de localisation a été validé sur une seule sphère augmentée. Une caméra monoculaire de résolution 800×600, cadencée à 45 Hz, est déplacée manuellement autour de la sphère avec des mouvements nécessitant l'estimation des 6 degrés de liberté de la caméra. La figure 6.4 montre la trajectoire obtenue ainsi que quelques positions clés. L'utilisation d'images sphériques permet de se localiser dans toutes les directions. La robustesse de l'algorithme de localisation permet également une localisation dans le voisinage de la sphère, jusqu'à 3.5 mètres de distance dans l'exemple proposé.

6.6.3.2 Graphe d'images sphériques

L'algorithme de localisation temps-réel a ensuite été validé sur un sous ensemble de 12 sphères augmentées extraites du graphe sphérique présenté dans les résultats de la section 4.4, pour localiser une caméra monoculaire de résolution 800×600, cadencée à 45 Hz et montée sur un véhicule de type Cycab naviguant localement à l'intérieur du graphe.

La figure 6.5, montre la trajectoire obtenue par l'algorithme de localisation : au point de départ, le véhicule est placé aux coordonnées $(X = -0.4, Z = -0.1)$, correspondant à l'image 6.5(a) et se dirige en direction des Z positifs, jusqu'au point $(X = -1.8, Z = 20.5)$. Le véhicule effectue ensuite une marche arrière (partie verte) jusqu'au point de coordonnées $(X = 3.3, Z = 18.8)$, puis retourne en sens inverse en direction du point de départ. On peut voir que l'utilisation d'images sphériques permet une localisation dans deux directions de navigation différentes en utilisant les mêmes sphères augmentées, pour des trajectoires localement différentes de l'apprentissage (jusqu'à 3 mètres de distance).

6.6.3.3 Graphe d'images perspectives

La méthode de localisation a également été validée sur des séquences d'apprentissage stéréo capturées en milieu urbain dans le cadre du projet CityVIP, dans le XIIeme arrondissement de Paris. Les séquences ont été acquises sur un véhicule naviguant normalement dans le flux de circulation (*i.e.* 50 km/h). Les caméras utilisées fournissent des images de résolution 800×600 pixels à une fréquence de 15 Hz. A cette fréquence les déplacements

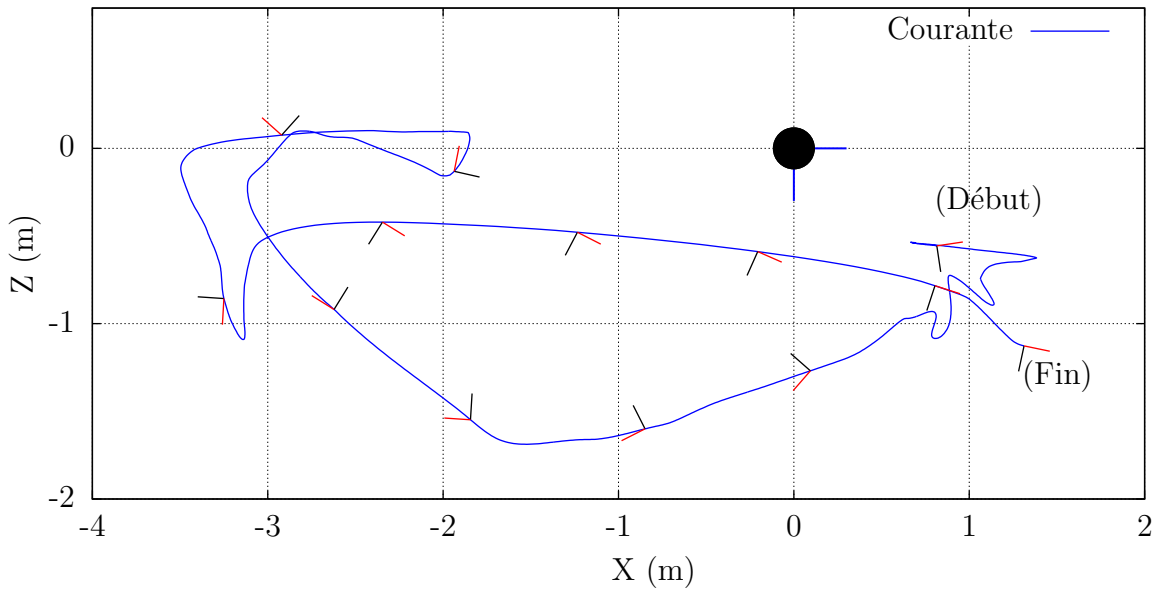


FIG. 6.4 – Localisation autour d'une image sphérique. En bleu : la trajectoire de la caméra estimée. Quelques poses sont affichées en rouge et noir, l'axe optique de la caméra étant représenté en rouge.

entre deux images successives peuvent être importants, notamment dans les virages pour les mouvement en rotations.

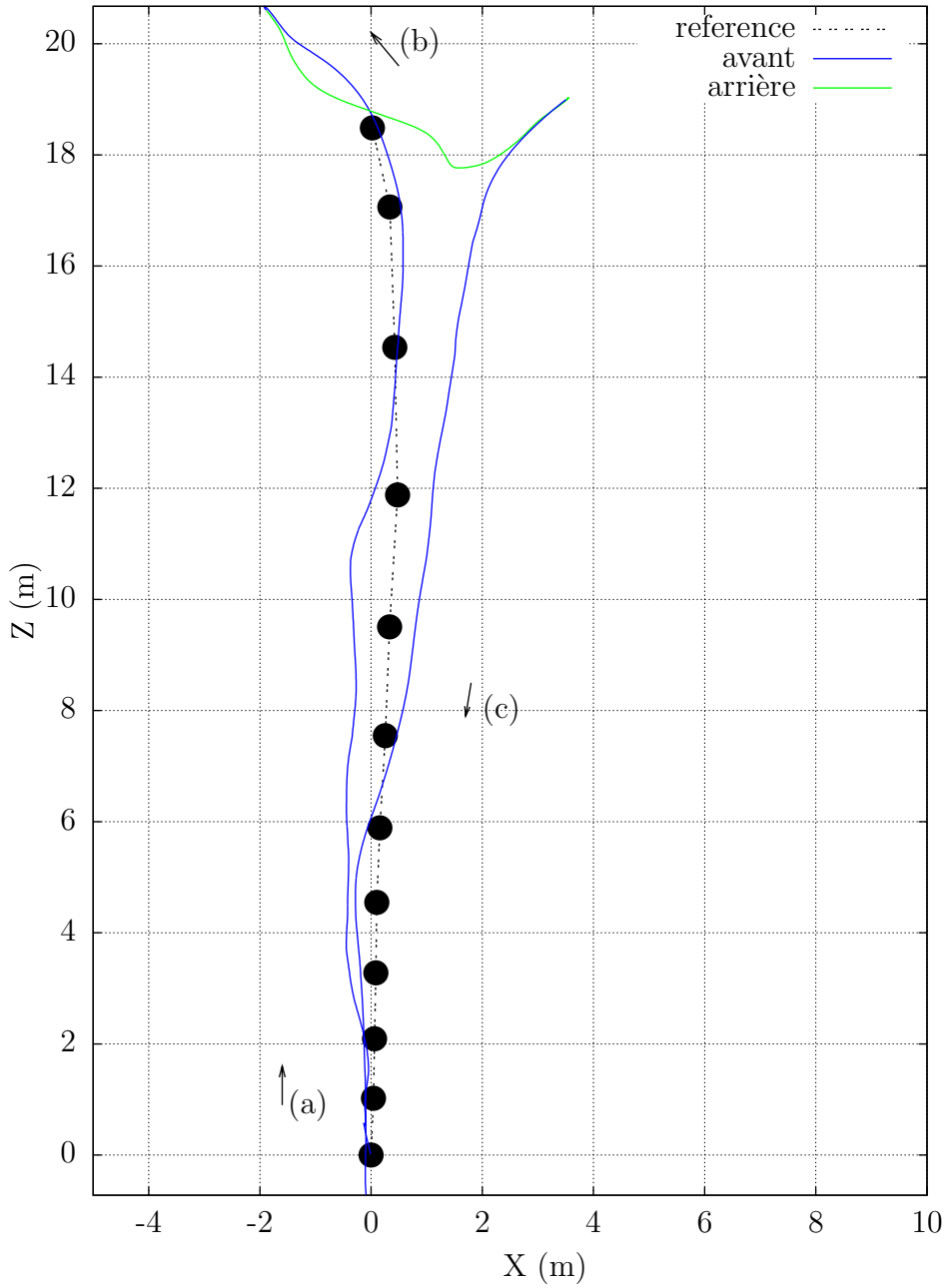
L'apprentissage a été effectué sur une première trajectoire d'environ 1 km, résultant en un graphe de 441 images de référence. Puisque seulement des images perspectives sont disponibles, un plus grand nombre d'images de référence est généré dans les virages afin d'assurer le recouvrement entre les nœuds du graphe (*cf.* figure 6.6).

Une localisation en ligne a été effectuée en utilisant une seconde séquence d'images, enregistrée lors d'un second passage sur une trajectoire légèrement différente avec les mêmes caméras (seulement une caméra est utilisée pour la localisation). La figure 6.6 montre la trajectoire obtenue ainsi que la position des images de référence. Bien que le véhicule est suivi le même chemin, la trajectoire est localement différente de celle d'apprentissage (dépassements, virages plus larges *etc*).

6.6.4 Environnement intérieur

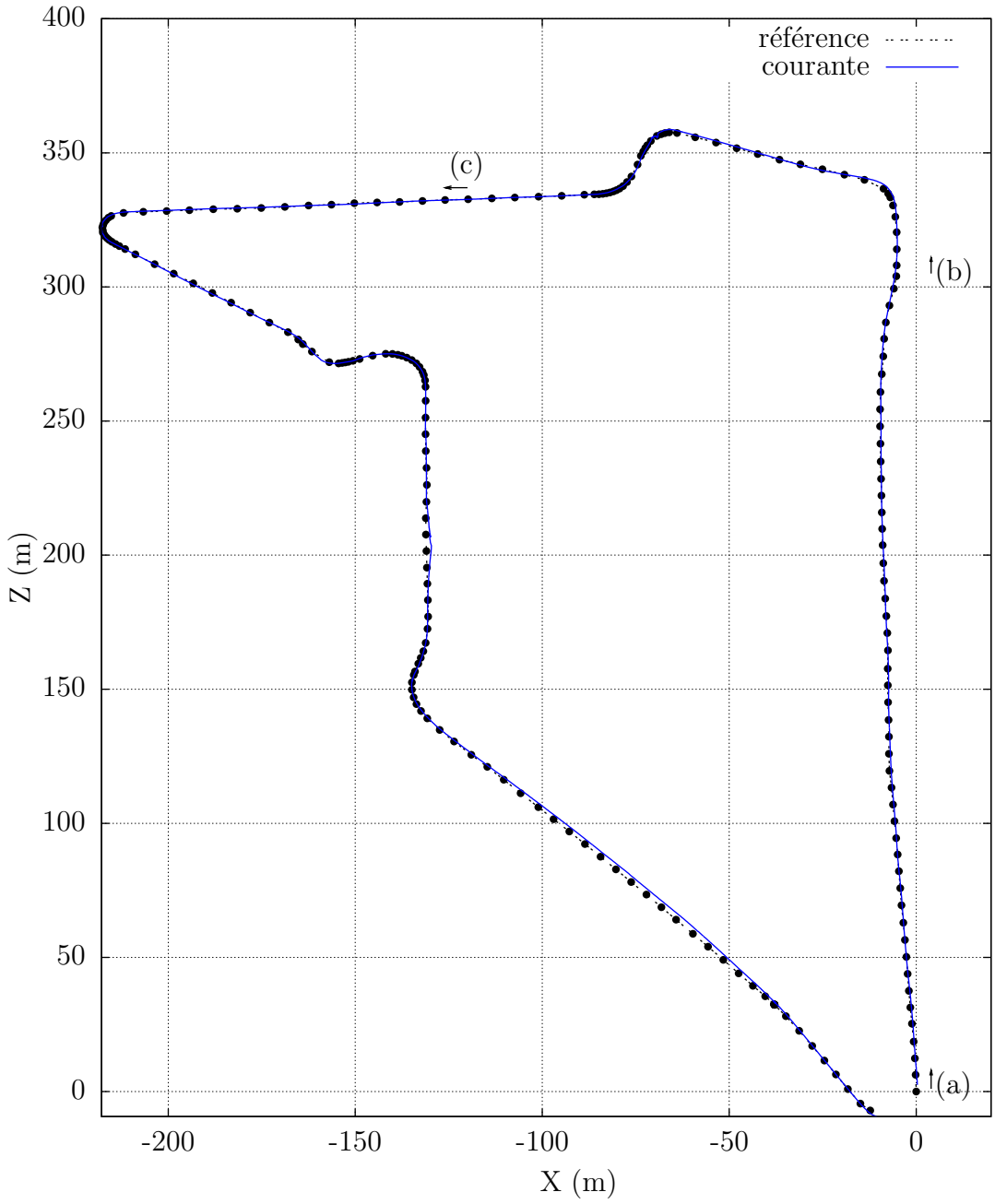
D'autres expérimentations ont également été réalisées dans un environnement intérieur, en utilisant une simple paire de caméras stéréo à *baseline* réduite (25cm) pour construire la base de données. Une caméra monoculaire est ensuite déplacée manuellement dans l'environnement, avec de fort mouvements à 6 degrés de liberté, permettant de valider la robustesse de l'algorithme.

Lors de l'apprentissage, la paire de caméras stéréo est déplacée manuellement de manière à cartographier localement la scène (halle robotique de l'INRIA Sophia Antipolis). Un graphe de 6 images augmentées est alors obtenu. Lors de la phase en ligne, la caméra monoculaire est manipulée dans la pièce, autour de la région d'apprentissage, avec des changements de points du vue importants. La figure 6.7(e), montre une reconstruction 3D de la scène, obtenue en ramenant dans le même repère les nuages de points 3D extraits des



(a) (b) (c)

FIG. 6.5 – Localisation dans un graphe d'images sphériques. En bleu : le véhicule navigue en marche avant. En vert : le véhicule effectue une marche arrière. (a),(b) et (c), des images capturées lors de la localisation en ligne.



(a) (b) (c)

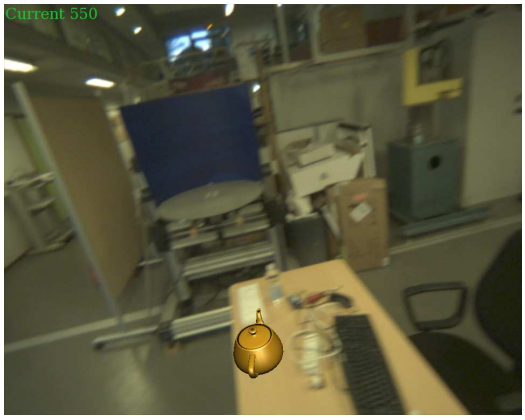
FIG. 6.6 – Localisation dans un graphe d'images perspectives dans le XIIeme arrondissement de Paris. En bleu : la trajectoire estimée par l'algorithme de localisation en ligne. En noir la position des images de référence (une image de référence sur 2 est affichée). (a),(b) et (c), des images capturées lors de la localisation en ligne.

images de référence. Les points utilisés pour la localisation en ligne apparaissent en orange dans l'image. La figure 6.7(a) montre l'image de référence utilisée pour la localisation de l'image courante 6.7(b). Lors du suivi, une image virtuelle du modèle est synthétisée en temps-réel à partir des points 3D et de la texture de l'image référence 6.7(c), à la position courante estimée. La carte de profondeur synthétisée 6.7, est alors utilisée pour augmenter l'image courante avec un objet virtuel (théière). Malgré le changement de point de vue important entre l'image courante et l'image de référence, l'algorithme est capable de fournir une localisation précise de la caméra courante (*cf.* trajectoire figure 6.8).

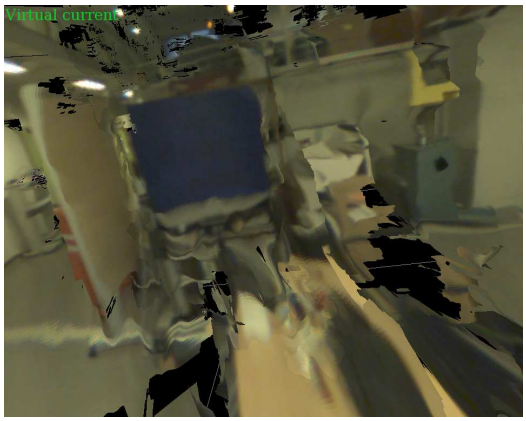
6.7 Conclusion

Dans ce chapitre, l'utilisation en ligne de la base de donnée d'images augmentées a été présentée. Une méthode directe, profitant des avantages de la représentation sphérique ego-centrée, permet la localisation précise d'une caméra, naviguant localement à l'intérieur du graphe. Afin d'obtenir des trajectoires lisses, plusieurs nœuds du graphe sont utilisés simultanément dans la minimisation, sans coût calculatoire supplémentaire grâce à la sélection de pixels saillants. Cette même sélection permet d'utiliser un nombre réduit de pixels, afin d'effectuer la localisation de la caméra à une fréquence de 45 Hz. Enfin, une technique directe est employée pour estimer les changements d'orientation rapide de la caméra. La robustesse et l'efficacité de l'algorithme permettent à la fois la localisation d'une caméra en environnement urbain et en environnement intérieur, sans utilisation d'autres capteurs ni de modèle prédictif. Les résultats expérimentaux illustrent les différents avantages d'un modèle ego-centré dense : l'estimation de pose est précise, robuste aux occultations et l'usage de sphères permet la localisation dans toutes les directions, dans un rayon assez large autour des images de référence.

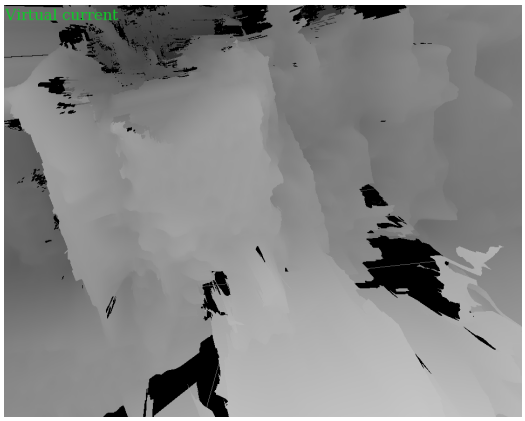
(a) Image référence.



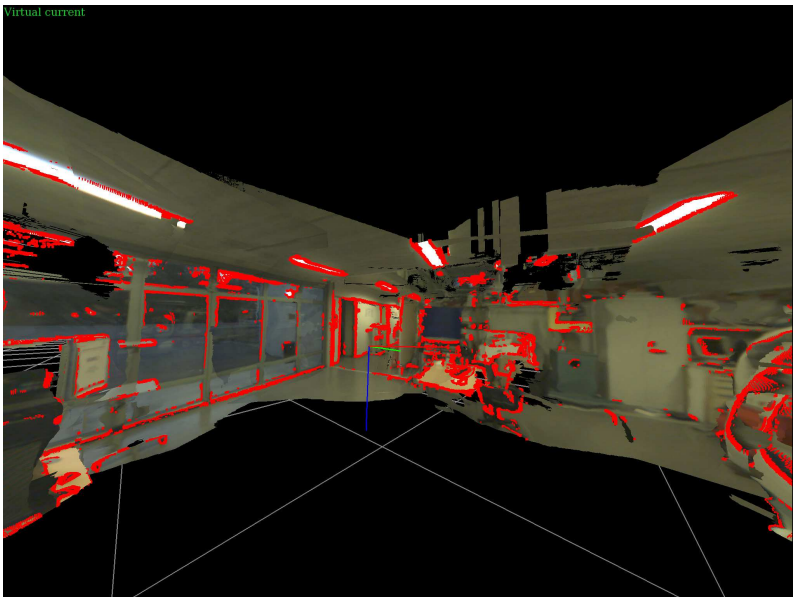
(b) Image courante augmentée.



(c) Image synthétisée.



(d) Profondeur synthétisée.



(e) Vue 3D synthétisée du modèle

FIG. 6.7 – Localisation en environnement intérieur. (a) : L'image de réference utilisée pour la localisation. (b) : L'image courante augmentée par un objet virtuel. (c) : Image synthétisée à la pose courante. (d) : Carte de profondeur synthétisée à la pose courante. (e) : Vue 3D du modèle synthétisée. En orange les pixels utilisés lors de la localisation.

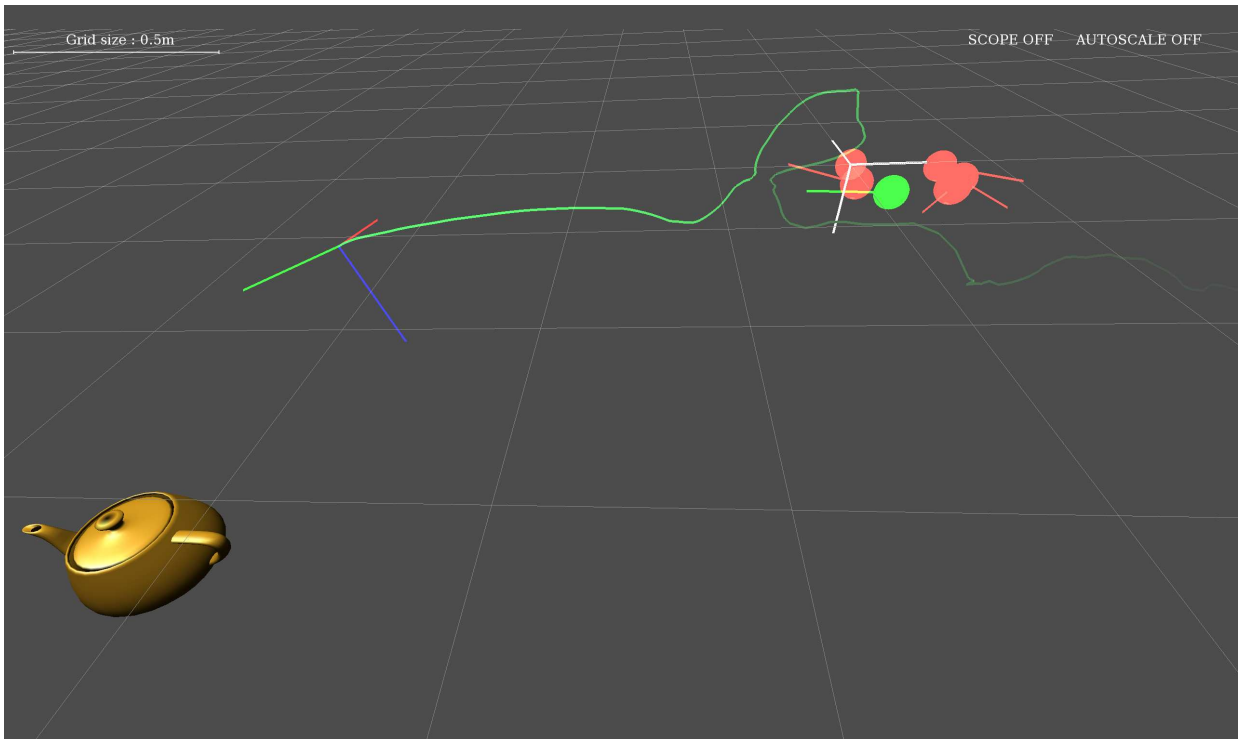


FIG. 6.8 – Trajectoire en environnement intérieur. La pose de l'image courante de la figure 6.7 est affichée par rapport à l'objet virtuel, et les images de référence affichées par les sphères rouges. La sphère verte est l'image de référence la plus proche.

Chapitre 7

Robustesse aux changements d'illumination

7.1 Introduction

Ce chapitre présente l'amélioration de la robustesse des techniques directes d'alignement d'images, en présence de variations d'illumination. En effet les conditions d'acquisition des images de référence augmentées du graphe, construit lors de la phase d'apprentissage, peuvent être très différentes de celles perçues lors de la phase de localisation en ligne :

- **Caméras différentes** : la réponse en intensité est différente d'une caméra à l'autre (optique, capteur CCD ou CMOS, capteur avec filtre de Bayer ou monochrome ...).
- **Réglages de la caméra** : ouverture, temps d'exposition, gain, correction gamma, effets de vignettage de l'objectif.
- **Illumination de la scène** : en fonction de la période de l'année ou de l'heure de la journée (position du soleil, nuages, lumière ambiante ...) l'apparence d'une même scène change considérablement.

Dans certains cas, des erreurs d'apparence trop importantes peuvent rendre difficile, voire impossible, un alignement direct des images, basé sur la minimisation des erreurs d'intensité (*cf.* chapitre 2).

La première section de cette partie présente un état de l'art des techniques directes robustes aux changements d'illumination, puis la seconde section présente une nouvelle technique hybride, robuste aux changements d'illuminations et aux occultations, en combinant un suivi basé modèle et un suivi basé odométrie visuelle. La méthode proposée est finalement validée par une implémentation temps réel, sur des séquences d'images contenant des variations d'illumination importantes.

7.2 État de l'art

Dans la littérature, plusieurs approches de suivi visuel direct robustes aux changements d'illumination ont été proposées, essentiellement pour du suivi de surfaces planaires, paramétrées par une homographie. Bartoli (2008) propose un modèle affine prenant seulement en compte les changements linéaires globaux dans les images. Dans Silveira & Malis (2007, 2010), un modèle local est proposé, en subdivisant la cible planaire en plusieurs patchs plans

contraints entre eux par une surface paramétrique. Cette approche apporte une robustesse aux changements locaux (réflexions spéculaires) et aux changements globaux. Cependant, la subdivision en patchs ajoute un nombre d'inconnues non négligeable, rendant l'algorithme complexe et difficilement exploitable en temps réel pour de grandes images. De plus la méthode n'est pas compatible avec des estimateurs robustes. Dans Hager & Belhumeur (1998), une méthode directe de suivi de visage, robuste aux changements d'illumination et aux occultations est présentée. Une représentation invariante de l'image est construite lors d'une phase d'apprentissage, à partir d'un ensemble d'images acquises sous différentes conditions d'illumination. Cependant cette phase d'apprentissage est délicate, et n'est pas robuste aux occultations ou aux ombres portées. Gonçalves & Comport (2011) proposent une technique basée sur les M-estimateurs et une mesure robuste du biais global. La technique proposée est robuste aux réflexions spéculaires (locales) et aux changements d'illumination globaux.

Au lieu d'utiliser une somme des carrés des différences (SSD) standard (*cf.* Baker & Matthews (2001)), d'autres fonctions de coûts plus robustes peuvent être utilisées. Dans Irani & Anandan (1998), une corrélation croisée normalisée (ZNCC) est appliquée sur des fenêtres locales. La technique présentée permet d'aligner des images très différentes, pour des translations bi-dimensionnelles. Une méthode robuste permet de rejeter les aberrations, mais nécessite de calculer le déterminant de la matrice Hessienne pour chaque pixel, ce qui est difficilement réalisable en temps réel. Plus récemment Panin & Knoll (2008); Dame & Marchand (2010) proposent de maximiser l'information partagée entre deux images, en utilisant l'information mutuelle (Viola & Wells (1995)). Déjà très utilisées dans le domaine médical pour l'alignement d'images (*e.g.* IRM, rayons-X), ces techniques sont actuellement les plus efficaces pour l'alignement d'images multi-modales (*e.g.* carte routière et image satellite). Cependant, elles nécessitent d'approximer une fonction de coût non continue afin de calculer un gradient analytique, ce qui peut dégrader la précision et rendre la convergence peu performante (lente) et même incertaine dans certains cas.

Récemment, Richa *et al.* (2011) ont proposé une approche basée sur la somme des variances conditionnelles (SCV), une mesure globale robuste aux changements d'illumination non linéaires. Cette métrique, similaire à l'information mutuelle est cependant très proche d'une SSD. En pratique, l'image de référence est incrémentalement mise à jour à la fin de chaque alignement. Pour cela les auteurs calculent l'histogramme conjoint entre l'image de référence et la dernière image recalée. La fonction d'intensité de l'image de référence est ensuite mise à jour en fonction de cet histogramme. Cependant, pour un calcul efficace des histogrammes, il est nécessaire de quantifier les valeurs images, ce qui peut réduire la précision et créer des minima locaux lors de l'alignement.

Dans le domaine de la *SfM* (*Structure from motion* Seitz *et al.* (2006)), des modèles d'illumination, physique (Cook & Torrance (1982)) ou empirique (Blinn (1977), plus complexes sont utilisés, afin d'estimer la normale des surfaces ainsi que leur coefficient d'albédo (*cf.* Zhang *et al.* (2003); Basri & Jacobs (2001)). En général les réflexions spéculaires ne sont pas prises en compte, et seulement un éclairage distant est considéré, ce qui n'est pas valide dans des environnements non contrôlés. De plus, ces techniques sont, la plupart du temps, utilisées hors-ligne et ne conviennent pas pour des approches temps réel.

Les techniques de suivi visuel temps réel utilisant des méthodes directes et considérant des changements d'illumination, abordent seulement le suivi de modèles géométriques basiques : plans Silveira & Malis (2007); Hager & Belhumeur (1998) ou cylindres Cascia & Sclaroff (1999) et seulement des régions de petite taille sont suivies (patchs). De plus,

pour les méthodes où les changements d'illumination font partie des inconnues du système, Silveira & Malis (2007); Bartoli (2008), les aberrations sont difficilement dissociables des changements d'illumination, il devient alors difficile d'être robuste simultanément à ces deux perturbations.

Dans nos travaux, des environnements 3D complexes sont considérés ainsi que des images de grande taille. De nombreux effets indésirables sont présents dans la scène : auto-occultations et auto-ombrage des objets, réflexions inter-objets, réflexions spéculaires. De plus, l'information géométrique obtenue par mise en correspondance dense stéréo, est souvent bruitée : des erreurs supplémentaires apparaissent lors des *warping* d'image, et il est difficile d'obtenir une bonne estimation des normales associées à chaque pixel. Il est donc nécessaire de définir un modèle flexible, robuste aux erreurs de modélisation géométriques et aux effets listés précédemment, mais le plus "léger" possible pour une utilisation temps réel.

7.3 Suivi hybride

7.3.1 Modèle d'illumination biais global, gain local

Dans un premier temps, les changements d'illumination présents entre une image acquise à un instant t et une autre image acquise à un instant $t+1$ peuvent être classés selon 4 types :

- **Échelon global** : Changement abrupt de l'intensité globale sur toute l'image : lumière ambiante allumée ou éteinte subitement, nuage masquant le soleil ...
- **Échelon local** : Changement abrupt de l'intensité locale dans l'image : lumière dirigée allumée ou éteinte subitement (projecteur), réflexions spéculaires, ombres et occultations causées par un changement de point de vue.
- **Rampe globale** : Changement lent de l'intensité globale de l'image : nuages, position du soleil...
- **Rampe locale** : Changement lent de l'intensité locale : lumière dirigée se déplaçant lentement.

Afin de prendre en compte les changements d'illumination globaux et locaux, la fonction de *warping* des intensités de l'équation 2.22 peut être reformulée par :

$$\boldsymbol{\mathcal{I}}^*(\mathbf{p}^*) = \boldsymbol{\alpha}\boldsymbol{\mathcal{I}}\Big(w(\mathbf{T}(\tilde{\mathbf{x}}); Z, \mathbf{p}^*)\Big) - \beta, \tag{7.1}$$

où $\beta \in \mathbb{R}$ est un biais global des intensités correspondant au changement global d'illumination sur toute l'image, et $\boldsymbol{\alpha} = diag(\alpha_1, \ldots, \alpha_i) \in \mathbb{R}^{mn \times mn}$ est une matrice diagonale correspondant au gain local affine par-pixel, l'image $\boldsymbol{\mathcal{I}}^*$ étant de dimensions $m \times n$ pixels.

Avec cette modélisation, le biais β permet de corriger les deux types de changements globaux d'une image : échelon et rampe. La matrice de gains alpha permet d'appliquer une correction affine locale, indépendante pour chaque pixels. Cela permet de corriger les deux autres types de changements locaux : échelon et rampe.

7.3.2 Suivi basé modèle

Le suivi basé modèle (MB) peut être défini comme un alignement 3D d'images classique entre un modèle acquis à l'instant $t_0 = 0$ et l'image courante acquise à l'instant $t = t_0 + \Delta_t$. Cette approche peut être considérée comme étant en permanence en configuration de type

échelon, puisque l'intervalle Δ_t entre l'acquisition du modèle et l'acquisition de l'image courante peut être important.

En utilisant la nouvelle fonction de l'équation (7.1), l'erreur à minimiser dans la fonction de coût devient :

$$\overline{\mathbf{e}(\mathbf{x})}_{MB} = \boldsymbol{\alpha}_{MB}\boldsymbol{\mathcal{I}}\left(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); Z, \mathbf{p}^*)\right) - \boldsymbol{\mathcal{I}}^*(\mathbf{p}^*) - \beta_{MB}, \tag{7.2}$$

Le changement d'illumination global β_{MB}, peut être déterminé de manière efficace tel que défini dans Gonçalves & Comport (2011), c'est à dire à chaque itération de la minimisation :

$$\beta_{MB} = Median\left(\mathbf{e}(\mathbf{x})_{MB}\right) \tag{7.3}$$

Ce biais, consiste en réalité à re-centrer la distribution de l'erreur autour de zéro, par la valeur de sa médiane, plus robuste aux valeurs aberrantes qu'une moyenne classique obtenue par exemple avec le modèle affine de Bartoli (2008).

La fonction de coût robuste, basée modèle, est alors définie par :

$$\mathcal{O}_\rho(\mathbf{x}) = \rho\left(\boldsymbol{\alpha}_{MB}\boldsymbol{\mathcal{I}}\left(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); Z, \mathbf{p}^*)\right) - \boldsymbol{\mathcal{I}}^*(\mathbf{p}^*) - \beta_{MB}\right), \tag{7.4}$$

Alors que le changement d'illumination global est pris en compte de manière efficace par le biais β_{MB}, le gain par-pixel $\boldsymbol{\alpha}_{MB}$ ne peut pas être observé localement, et n'est pas dissociable des occultations. Cependant, ces changements locaux sont absorbés par la matrice de pondération $\mathbf{D}_{MB}$ obtenue lors de la minimisation itérative robuste (*cf.* section 2.6.5).

Bien que robuste à des changements d'illumination non négligeables, une pondération excessive des erreurs ralentit la vitesse de convergence de l'algorithme de recalage et crée des minima locaux. Pour un algorithme temps-réel, le nombre d'itérations maximales possible étant limité, la méthode de localisation peut échouer.

7.3.3 Suivi basé odométrie visuelle

Dans cette section, une technique basée odométrie visuelle (VO) monoculaire, à 6 degrés de liberté est présentée. Puisque les images augmentées fournissent l'information géométrique de la scène, il est possible de formuler le problème de localisation en utilisant la géométrie du modèle ($\boldsymbol{\mathcal{Z}}^*$) et l'image $\boldsymbol{\mathcal{I}}_{t-1}$ acquise à l'instant $t-1$, recalée sur l'image de référence par :

$$\boldsymbol{\mathcal{I}}^w_{t-1}(\mathbf{p}^*) = \boldsymbol{\mathcal{I}}_{t-1}(w(\mathbf{T}_{t-1}, Z, \mathbf{p}^*)). \tag{7.5}$$

Cette image $\boldsymbol{\mathcal{I}}^w_{t-1}$, peut être considérée comme une nouvelle image de référence (*cf.* figure 7.1), ce qui permet de définir une nouvelle erreur basée odométrie visuelle, définie comme l'erreur entre deux images acquises successivement :

$$\overline{\mathbf{e}(\mathbf{x})}_{VO} = \boldsymbol{\alpha}_{VO}\boldsymbol{\mathcal{I}}_t(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); Z, \mathbf{p}^*)) - \boldsymbol{\mathcal{I}}^w_{t-1}(\mathbf{p}^*) - \beta_{VO}, \tag{7.6}$$

où comme pour le suivi basé modèle, les changements d'illumination sont modélisés par un biais global β_{VO} et un gain local $\boldsymbol{\alpha}_{VO}$. Cette erreur peut être minimisée de manière classique (*cf.* section 2.6.5), afin d'estimer la transformation $\mathbf{T}(\mathbf{x})$.

Contrairement aux techniques d'odométrie visuelle classiques Comport *et al.* (2010), cette méthode évite l'estimation coûteuse de la structure de la scène, habituellement obtenue

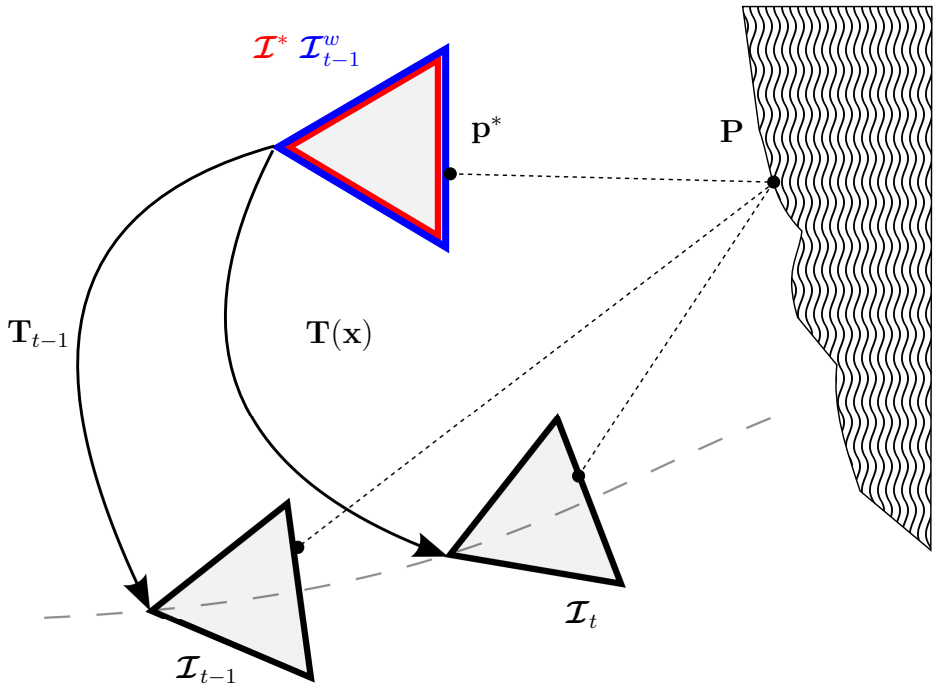


FIG. 7.1 – Position des caméras dans la scène. En rouge, l'image de référence augmentée $\mathcal{I}^*$, en bleu l'image $\mathcal{I}^w_{t-1}$ recalée sur le modèle.

par mise en correspondance dense stéréo, en s'appuyant sur la géométrie déjà présente dans le modèle de référence. Cela permet d'une part d'effectuer une localisation à 6 degrés de liberté avec une caméra monoculaire, et d'autre de permettre un calcul très rapide de la pose en évitant la reconstruction en ligne.

Si le flux d'images considéré est acquis à la fréquence d'une caméra (*e.g.* ≥ 25 Hz), les changements d'illumination entre deux images successives sont beaucoup moins importants que dans le cas d'un suivi basé modèle : $\boldsymbol{\alpha}_{VO} \approx 1$ et $\beta_{VO} \ll \beta_{MB}$. Dans ce cas, la méthode basée VO est robuste aux changements d'illumination locaux de type rampe, mais aussi aux changements globaux grâce à la mesure du biais (*cf.* figure 7.2).

Cependant comme toute méthode d'odométrie visuelle, les erreurs d'estimation de poses, liées aux erreurs d'interpolation, aux erreurs géométriques du modèle et aux critères d'arrêt de la minimisation, sont intégrées dans le temps, introduisant une dérive dans la localisation.

7.3.4 Optimisation globale

Dans cette section, une approche hybride est présentée, combinant les avantages du suivi basé modèle et du suivi basé odométrie visuelle. La première méthode, le suivi MB, maintient une localisation par rapport à un modèle fixe, ce qui permet de ne pas dériver. Cependant, cette approche est soumise à des changements d'illumination très importants, résultant en une convergence plus lente. Le suivi VO quant à lui, permet d'utiliser un modèle de référence et une image courante temporellement proches, ce qui conduit à une convergence rapide, mais dérive au cours du temps.

Pour une technique directe de localisation, il est crucial de maintenir une mesure basée capteur dans le procédé de minimisation, pour conserver un maximum de précision et éviter d'intégrer les erreurs. Dans ce cas il n'est pas idéal de mettre à jour incrémentalement l'image

de référence tel que dans Richa *et al.* (2011).

La méthode hybride proposée consiste à définir une erreur globale $\overline{\mathbf{e}(\mathbf{x})}_H$, basée sur un empilement de l'erreur VO et l'erreur MB :

$$\overline{\mathbf{e}(\mathbf{x})}_H = \begin{bmatrix} \rho\big(\boldsymbol{\mathcal{I}}_t(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); Z, \mathbf{p}^*) - \boldsymbol{\mathcal{I}}^*(\mathbf{p}^*) - \beta_{MB}\big) \\ \rho\big(\boldsymbol{\mathcal{I}}_t(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); Z, \mathbf{p}^*) - \boldsymbol{\mathcal{I}}^w_{t-1}(\mathbf{p}^*) - \beta_{VO}\big) \end{bmatrix}. \tag{7.7}$$

La matrice Jacobienne complète, associée au vecteur d'erreur $\overline{\mathbf{e}(\mathbf{x})}_H$ est alors définie par :

$$\mathbf{J}_H = \begin{bmatrix} \mathbf{J}_{\boldsymbol{\mathcal{I}}^*}\mathbf{J}_w\mathbf{J}_x \\ \mathbf{J}_{\boldsymbol{\mathcal{I}}^w_{t-1}}\mathbf{J}_w\mathbf{J}_x \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{MB} \\ \mathbf{J}_{VO} \end{bmatrix}. \tag{7.8}$$

Puisque la géométrie du modèle est partagée entre la partie VO et la partie MB, la nouvelle matrice $\mathbf{J}_H$ peut être obtenue efficacement (*cf.* équation (7.8)), car il est seulement nécessaire de mettre à jour la dérivée spatiale $\mathbf{J}_{\boldsymbol{\mathcal{I}}^w_{t-1}} = \nabla_{\mathbf{p}^w}(\boldsymbol{\mathcal{I}}^w_{t-1})$, associée à l'image recalée à l'instant $t-1$.

En général, les deux distributions des erreurs $\overline{\mathbf{e}(\mathbf{x})}_{MB}$ et $\overline{\mathbf{e}(\mathbf{x})}_{VO}$, peuvent appartenir à deux modalités différentes (*cf.* figure 7.3). Les deux mesures de biais β_{MB} et β_{VO} permettent de centrer de manière indépendante les erreurs correspondantes, et de calculer deux matrices de pondération $\mathbf{D}_{MB}$ et $\mathbf{D}_{VO}$ (*cf.* section 2.6.5), afin de construire une matrice de pondération (diagonale) globale de dimensions $2mn \times 2mn$:

$$\mathbf{D}_H = \begin{bmatrix} \mathbf{D}_{MB} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{VO} \end{bmatrix}. \tag{7.9}$$

L'incrément $\mathbf{x}$ est alors obtenu classiquement de manière robuste par :

$$\mathbf{x} = -\left(\mathbf{J}_H^T\mathbf{D}_H\mathbf{J}_H\right)^{-1}\mathbf{J}_H^T\mathbf{D}_H\overline{\mathbf{e}(\mathbf{x})}_H, \tag{7.10}$$

afin de mettre à jour incrémentalement la pose :

$$\widehat{\mathbf{T}} \leftarrow \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}). \tag{7.11}$$

L'algorithme 7.1, montre le déroulement de la méthode hybride, l'ajout du suivi basée *VO* ne demande que très peu de calculs supplémentaires.

7.3.5 Exemple de localisation hybride

La figure 7.2, montre un exemple d'images impliquées dans la méthode hybride après alignement, sur une séquence urbaine. Bien que l'intervalle Δ_t entre l'acquisition du modèle et l'acquisition de l'image soit faible ($\approx 10min$), la scène contient de nombreux changements d'illumination globaux et locaux ainsi que des occultations assez classiques en environnement urbain (*e.g.* objets n'appartenant pas au modèle : véhicules).

Après alignement, l'erreur basée modèle ($\mathbf{e}(\tilde{\mathbf{x}})_{MB}$) est très importante, la façade du bâtiment sur la gauche est saturée par le soleil dans l'image courante $\boldsymbol{\mathcal{I}}_t$, mais renvoie une lumière diffuse sur l'image de référence $\boldsymbol{\mathcal{I}}^*$. Des erreurs sont également générées par les véhicules et le cycliste non présents dans l'image de référence, ainsi que par les feuilles des arbres, localement saturées par la lumière du soleil. La distribution de l'erreur résultante est très "aplatie" (*cf.* histogramme rouge, figure 7.3), la matrice $\mathbf{D}_{MB}$ pondère fortement l'erreur. En revanche, l'erreur basée VO est très faible, puisque les changements d'illumination entre les images $\boldsymbol{\mathcal{I}}_{t-1}$ et $\boldsymbol{\mathcal{I}}_t$ sont quasiment nuls. Dans ce cas, seuls les objets mobiles (*e.g.* cycliste) sont rejetés par les M-estimateurs, permettant une convergence rapide vers la solution.

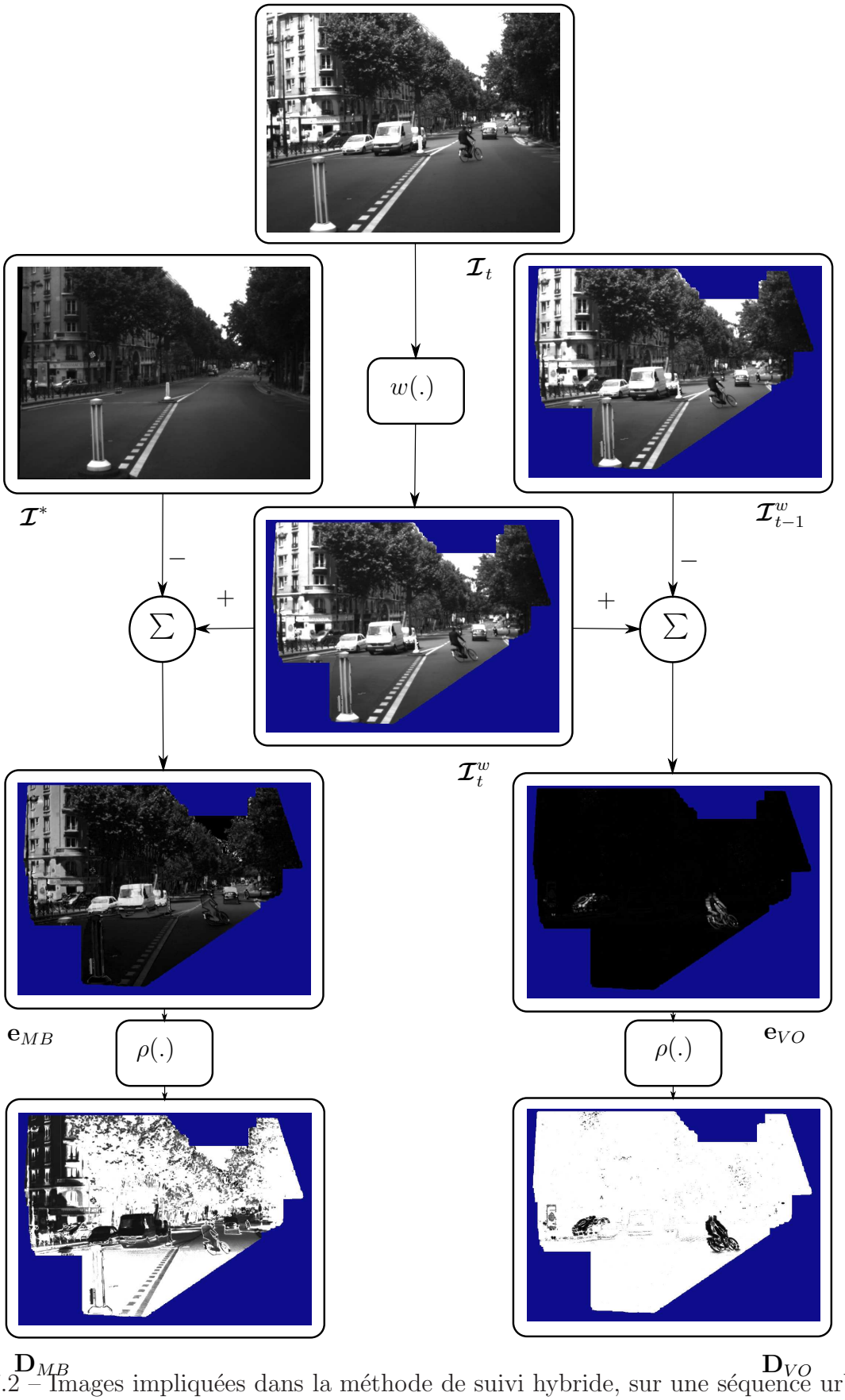


FIG. 7.2 – Images impliquées dans la méthode de suivi hybride, sur une séquence urbaine. En bleu, l'information de profondeur n'est pas disponible dans l'image de référence. La partie MB (gauche) est en permanence soumise à de forts changements d'illumination. La partie VO (droite), minimise une erreur entre deux images temporellement proches.

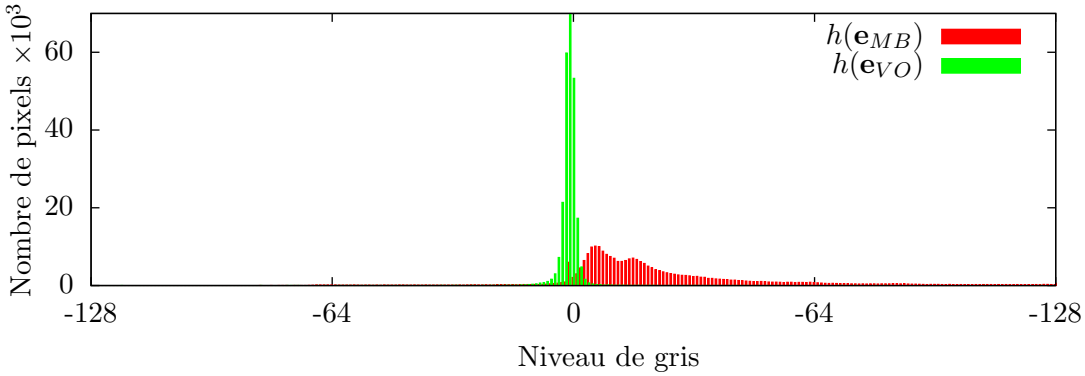


FIG. 7.3 – Histogrammes des erreurs MB et VO des images de la figure 7.2 : La distribution de l'erreur MB est très aplatie et non centrée à cause des changements d'illumination et des occultations. La distribution de l'erreur VO est comme attendue centrée en 0 et d'écart type très faible.

Algorithme 7.1 Algorithme de suivi hybride.

pour chaque image $\mathcal{I}_t$ **faire**
  calculer le gradient spatial de l'image $\mathcal{I}^w_{t-1}$.
  calculer la matrice Jacobienne $\mathbf{J}_{VO}$ (eq. 7.8).
  itération $\leftarrow 1$
  répéter
    calculer l'image transformée $\mathcal{I}^w_t = \mathcal{I}_t(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); Z, \mathbf{p}^*))$ (eq. 7.7).
    calculer les erreurs $\mathbf{e}(\mathbf{x})_{MB}$ et $\mathbf{e}(\mathbf{x})_{VO}$ (eq. 7.7).
    calculer les biais β_{MB} et β_{VO} (eq. 7.3).
    calculer les erreurs centrées $\overline{\mathbf{e}(\mathbf{x})}_{MB}$ et $\overline{\mathbf{e}(\mathbf{x})}_{VO}$ (eq. 7.3).
    calculer les matrices de pondération $\mathbf{D}_{MB}$ et $\mathbf{D}_{VO}$ (eq. 7.7).
    calculer l'incrément $\mathbf{x}$ (eq. 7.10).
    mettre à jour la pose : $\widehat{\mathbf{T}} \leftarrow \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x})$
    itération $\leftarrow$ itération $+1$
  jusqu'à $\|\mathbf{x}\| \leq \epsilon$ ou itération $>$ itération maximum
  mettre à jour modèle VO : $\mathcal{I}^w_{t-1} = \mathcal{I}^w_t$
fin pour

7.4 Résultats expérimentaux

7.4.1 Comparaison des techniques

Afin de valider les avantages de la technique hybride, un recalage direct à été effectué entre une image de référence augmentée et une image courante contenant d'importants changements d'illumination. Les trois fonctions de coût ont étés utilisées. La figure 7.4, montre la décroissance de l'erreur $\mathbf{e}_{[\cdot]}$ en fonction du nombre d'itérations. La valeur de l'erreur à été normalisée entre 0 et 1 pour chaque méthode, 0 indiquant que le minimum global de la fonction de coût est atteint, 1 étant la valeur à l'initialisation. Pour le suivi MB, il faut environ 22 itérations pour atteindre le minimum global de la fonction. Pour le suivi VO, la convergence est la plus rapide, seulement 10 itérations sont nécessaires. Enfin la technique hybride, combinant les deux précédentes techniques, a convergé en 12 itérations.

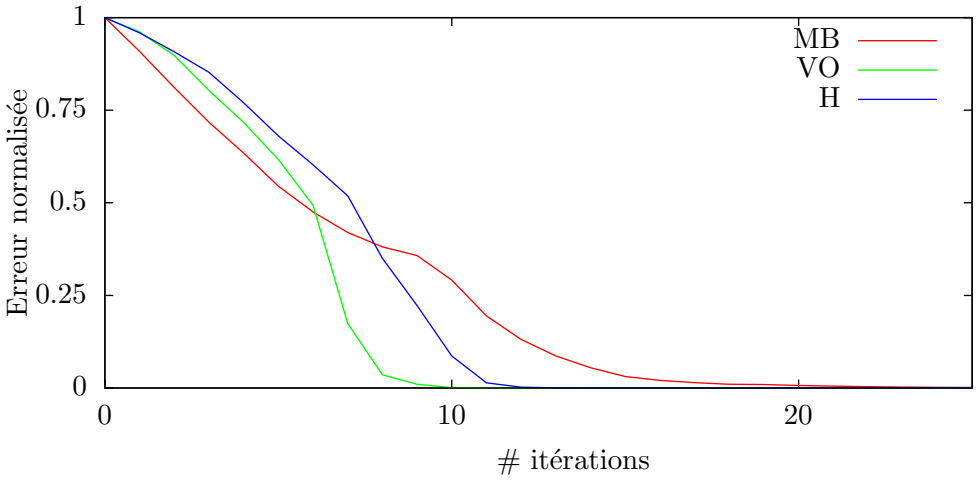


Fig. 7.4 – Vitesse de convergence des techniques MB,VO et H. La technique VO converge rapidement vers son minimum. A cause des changements d'illumination importants, la technique MB est plus lente. La technique hybride permet une convergence rapide tout en évitant la dérive.

Bien que la méthode VO semble plus efficace, une seconde expérience a été réalisée afin de mettre en évidence l'intérêt de la méthode hybride par rapport à la méthode VO. Un graphe d'images augmentées a été reconstruit à l'aide d'une paire de caméras stéréo. Une séquence d'images a ensuite été acquise en utilisant une caméra monoculaire. Des changements d'illumination ont été introduits à l'aide d'un projecteur afin de simuler des changements locaux et globaux ainsi que des ombres portées.

A l'instant t=0, la pose de la caméra courante par rapport à l'image de référence est initialisée à une position connue tel que $\mathbf{T}(\mathbf{x})|_{t=0} = \mathbf{I}$: pour respecter cette contrainte, la caméra est placée sur un trépied. La caméra courante est ensuite déplacée manuellement à l'intérieur du modèle de manière à générer des mouvements rapides suivant les 6 degrés de liberté. A la fin de la séquence, la caméra est replacée à sa position initiale (trépied), c'est à dire $\mathbf{T}(\mathbf{x})|_{t=N} = \mathbf{I}$. La séquence obtenue contient 1300 images, ce qui correspond à une durée d'environ 30 secondes. La pose de la caméra à tout d'abord été estimée avec la technique MB, ensuite avec la technique VO et enfin avec la technique hybride. Pour chaque

suivi, les mêmes paramètres d'optimisation ont étés utilisés (*i.e.* critères d'arrêt : nombre maximum d'itérations et seuil de convergence). La figure 7.5 illustre les résultats de cette expérience : pour les trois techniques, les valeurs de la norme en rotation et en translation de la pose estimée, sont affichées en fonction du temps. Pour la méthode MB, le suivi échoue à partir de l'image 761 : l'algorithme a divergé. La technique d'odométrie visuelle, a permis d'estimer la pose de la caméra sur toute la séquence. Cependant les erreurs successives d'alignement sont intégrées au cours du temps : la pose finale est erronée, $\mathbf{T}(\mathbf{x})|_{t=N} >> \mathbf{I}$. La méthode proposée, a permis de suivre la pose de la caméra sur toute la séquence. Puisque les intensités initiales de l'image de référence sont toujours prises en compte dans la fonction de coût, aucune dérive n'est intégrée : la pose finale correspond bien à l'identité.

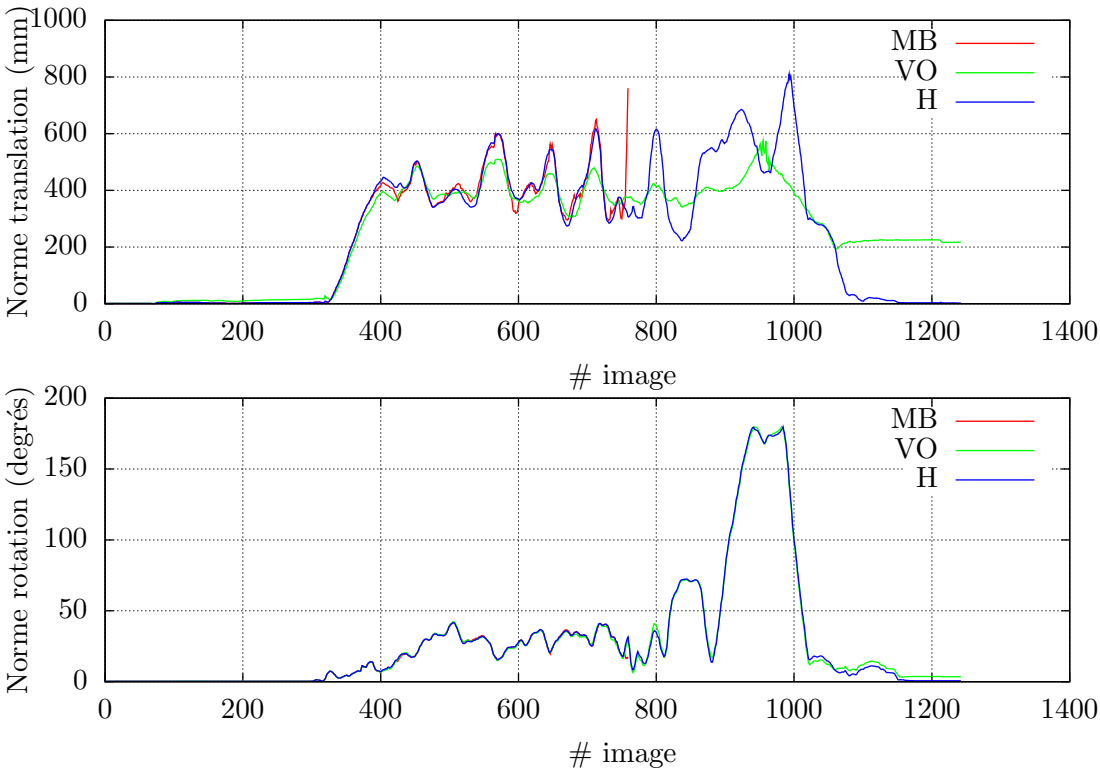


FIG. 7.5 – Mise en évidence de la dérive de l'odométrie visuelle : Haut, norme des valeurs en translation des poses estimées. Bas, norme des valeurs en rotation des poses estimées. La technique MB échoue à l'image 761. La technique VO dérive (la translation finale doit être égale à zéro). La technique H permet un suivi robuste sans dérive.

7.4.2 Robustesse

Afin de valider la robustesse de la méthode, une base de données d'images augmentées est construite à partir d'une paire de caméras stéréo. Le placement des nœuds du graphe et la sélection des images augmentées sont obtenus en utilisant la technique d'odométrie visuelle directe présentée en section 4.2, appliquée au cas d'un modèle de projection perspective. Le graphe obtenu contient 20 images augmentées de dimensions 800×600 cartographiant un environnement intérieur. La base de données est ensuite utilisée pour localiser en temps réel une caméra monoculaire cadencée à 45 Hz, fournissant des images de dimensions 800×600 pixels, déplacée manuellement dans l'environnement (avec des mouvements rapides). Afin

de générer des saturations dans les images, le diaphragme de la caméra a été volontairement ouvert au maximum. Plusieurs suivis ont été réalisés sous différentes conditions d'illumination (matin, jour et nuit) :

- Changement de l'intensité globale.
- Caméra saturée localement par des réflexions spéculaires (7.6(b)).
- Éclairage artificiel avec un projecteur en mouvement (7.8(b)).

A l'instant $t = 0$, il est nécessaire d'initialiser la position de la caméra courante par rapport au modèle. Tout d'abord, une recherche exhaustive du nœud du graphe le plus proche est effectuée en sélectionnant le meilleur score de corrélation après recalage sur toutes les images de la base de donnée (*cf.* section 6.6.2). Ensuite afin d'assurer un maximum de précision sur la première pose, un recalage dense basé modèle est effectué entre l'image de référence et la première image courante. Pour garantir convergence et précision, tous les pixels de l'image de référence sont utilisés et un grand nombre d'itérations est autorisé. Puisque c'est une phase d'initialisation, le temps de calcul n'est pas important et la caméra est maintenue statique pour quelques secondes. Une fois la première pose estimée, le suivi hybride peut être utilisé.

Les figures 7.6, 7.7 et 7.8 montrent des résultats d'alignement capturés en temps réel lors du suivi de la caméra. Chaque colonne représente un résultat d'alignement. La ligne (a) représente l'image de référence utilisée dans le suivi, la ligne (b) l'image courante perçue par la caméra, et enfin la ligne (c) représente une image de synthèse calculée à partir des points 3D, et de la texture de l'image de référence, rendue à la pose courante estimée. Cette image de synthèse doit donc correspondre visuellement à l'image courante.

On peut voir que la méthode proposée permet un suivi dans des conditions d'illumination assez extrêmes : fortes saturations locales et sous-exposition, changement d'illumination global, ombres portées et occultations. De plus, le suivi basé odométrie visuelle, minimisant directement une erreur inter-images, rend également l'algorithme très robuste au flou de bougé et aux dé-focalisations de la caméra (*cf.* figure 7.7).

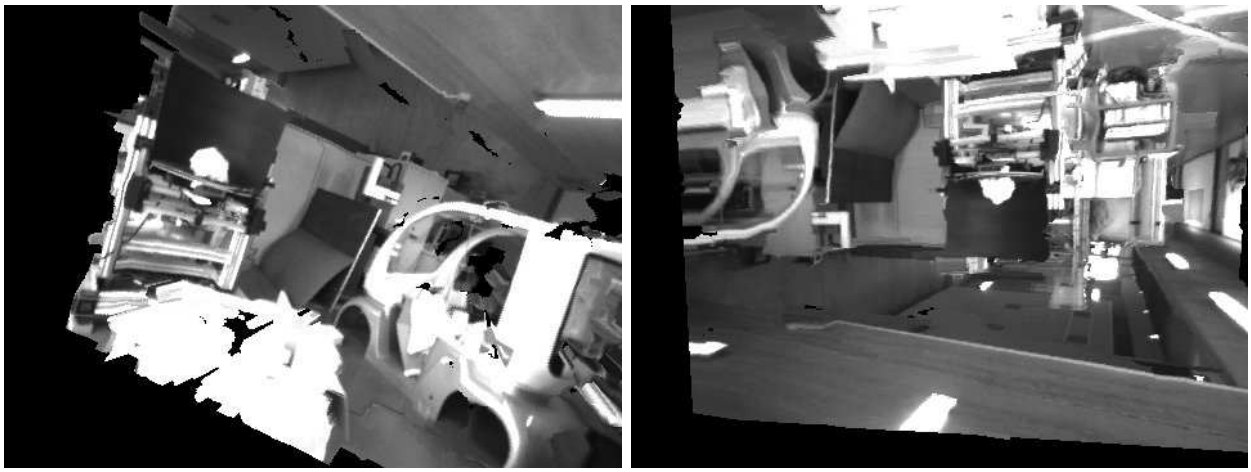
7.5 Conclusion

Dans la littérature, les approches de suivi visuel directe robustes aux changements d'illumination, abordent seulement le suivi de modèles géométriques basiques (plans,cylindres, *etc*). La nouvelle méthode de suivi hybride proposée dans ce chapitre, permet d'améliorer la robustesse aux changements d'illumination des méthodes directes de suivi visuel 3D, pour un faible coût calculatoire. Le fait d'utiliser un modèle d'illumination simple permet de traiter de manière efficace des environnements 3D complexes, en combinant une approche basée modèle et une approche basée odométrie visuelle. L'approche basée modèle permet une localisation précise sans dérive par rapport aux images du graphe de la base donnée, alors que l'approche basée odométrie visuelle assure la robustesse du système face à des changements d'illuminations locaux importants.

L'avantage de cette approche est tout d'abord qu'aucun modèle géométrique approximatif de la scène n'est nécessaire, ce qui évite par exemple l'estimation coûteuse des normales des pixels. Les expérimentations présentées dans la dernière section montrent la robustesse de l'algorithme face à plusieurs types de changement d'illumination, ainsi qu'aux occultations grâce aux M-estimateurs. Il faut enfin noter que la méthode proposée aborde le cas du suivi de scènes 3D, mais l'algorithme est applicable à d'autres types de suivis paramétriques (surfaces planaires, cylindres ...).

(a) Images référence

(b) Images courante

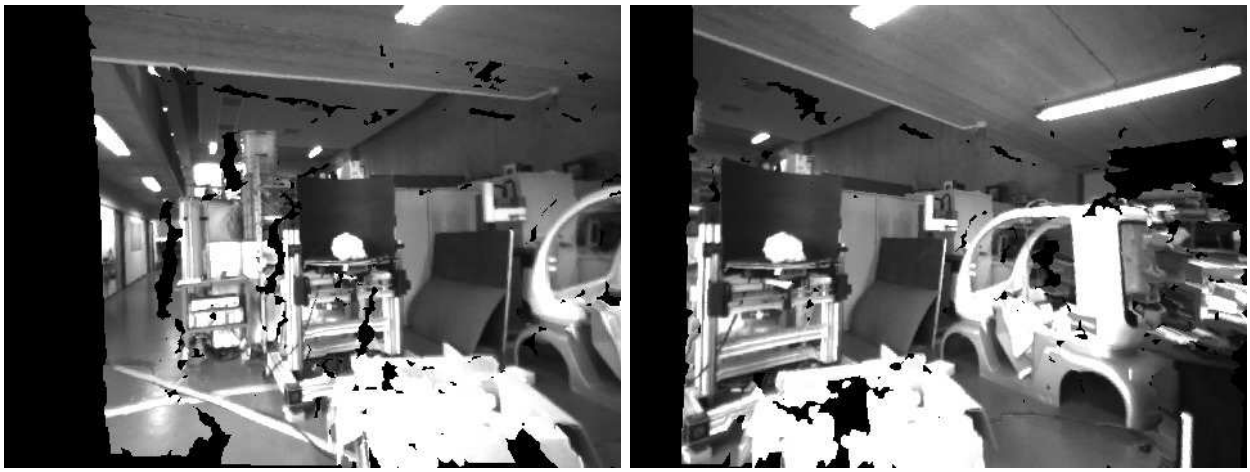


(c) Images virtuelles associées

FIG. 7.6 – A gauche, large mouvement à 6 degrés de liberté entre l'image de référence et l'image courante. A droite, rotation de 180 degrés suivant l'axe optique. Dans les deux cas des changements d'illumination globaux et locaux sont présents : saturations, lumières éteintes (tubes fluorescents).

(a) Images référence

(b) Images courante



(c) Images virtuelles associées

FIG. 7.7 – A gauche, l'image courante est dé-focalisée. A droite, le mouvement rapide de la caméra génère du flou de bougé. Dans les deux cas, des problèmes d'exposition sont présents dans les images (zones saturées ou sous-exposées).

(a) Images référence

(b) Images courante

(c) Images virtuelles associées

FIG. 7.8 – A gauche, lumière artificielle projetée (en mouvement), à droite, même expérience avec occultations. Dans les deux cas, l'image courante contient des ombres portées ainsi que des zones saturées.

Chapitre 8

Navigation autonome

8.1 Présentation du système

Dans cette partie, la méthode de localisation visuelle est utilisée pour une application de navigation autonome en environnement urbain. Cette application a été dévellopée dans le cadre des expérimentations finales du projet ANR CityVIP. L'objectif est d'utiliser la méthode de localisation visuelle pour suivre une trajectoire 3D prédéfinie. Un véhicule électrique de type Cycab (*cf.* figures 8.1 et 8.2), est équipé d'une caméra monoculaire Stingray F125B, montée sur la galerie du véhicule. La caméra est réglée à une fréquence de 45 Hz et sur une résolution de 800×600 pixels.

Une caméra IP a également été utilisée pour retransmettre une image vers une station déportée, permettant de suivre à distance la position du Cycab.

A l'avant du véhicule, un laser est utilisé pour la détection d'obstacles éventuels tel que des piétons ou d'autres véhicules mobiles. Pour effectuer tous les calculs nécessaires, c'est à dire la localisation, la détection d'obstacles et le contrôle du véhicule, un ordinateur portable équipé d'un processeur Intel core i7-2820QM est embarqué dans le véhicule. Pour la synchronisation des capteurs, l'architecture logicielle AROCCAM a été utilisée. Cette librairie permet de simplifier l'intégration d'algorithmes multi-capteurs, à travers une gestion événementielle des données.

8.2 Suivi de trajectoire

8.2.1 Modèle cinématique du véhicule

Un véhicule de type Cycab peut être modélisé par un modèle de type tricycle, illustré sur la figure 8.3, et définit selon le modèle cinématique suivant :

$$\begin{cases} \dot{x} = U\cos(\psi) \\ \dot{y} = U\sin(\psi) \\ \dot{\psi} = U/L\tan(\delta) \end{cases}, \tag{8.1}$$

où les coordonnées (x, y) sont les coordonnées Cartésiennes, U correspond à la vitesse longitudinale du véhicule, $\dot{\psi}$ est la vitesse angulaire, ψ est le cap, δ est l'angle de braquage des roues et L est la distance entre les axes des roues.

FIG. 8.1 – Véhicule Cycab et ses capteurs. La caméra Firewire est placée sur la galerie du véhicule. Le laser, positionné à l'avant, est utilisé pour détecter les obstacles à proximité du véhicule.

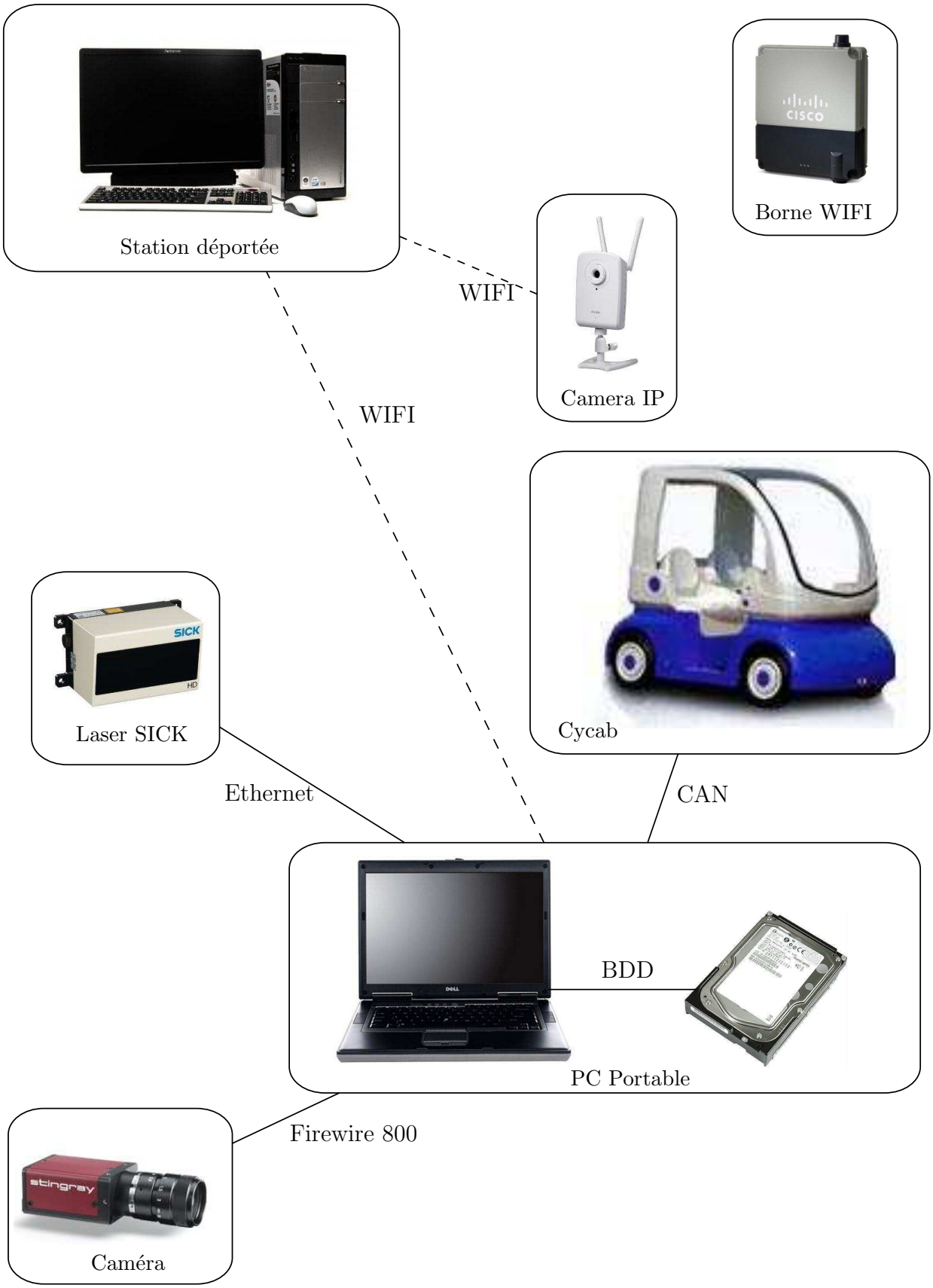


FIG. 8.2 – Système de navigation autonome. L'ordinateur portable assure la localisation et le contrôle du Cycab. Il communique avec un laser SICK part port Ethernet et avec la caméra par port Firewire. La station déportée permet de suivre à distance, les données noyées par la caméra IP et par l'algorithme de localisation.

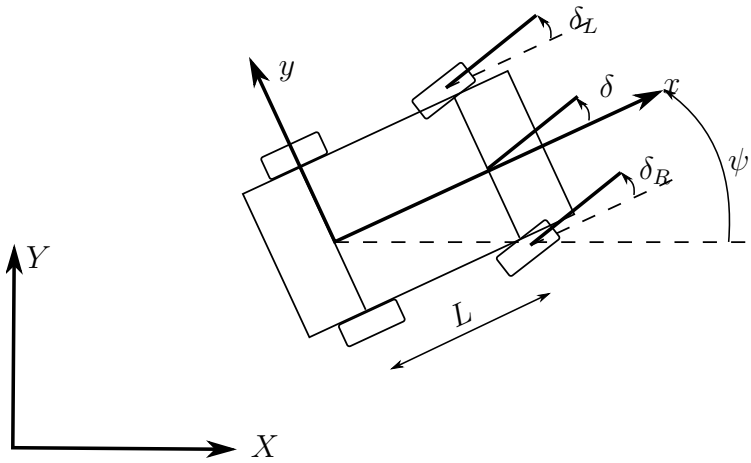


FIG. 8.3 – Modèle cinématique du véhicule.

8.2.2 Trajectoire de consigne

Tout d'abord un apprentissage est effectué avec le système de sphères visuelles augmentées. Le véhicule est conduit manuellement le long d'une trajectoire d'apprentissage afin de générer la base de donnée nécessaire à la localisation. Cette base de donnée est alors embarquée sur le véhicule utilisé pour la localisation en ligne.

L'objectif est de suivre automatiquement une trajectoire, générée dans le voisinage de celle d'apprentissage. Cependant, pour assurer une trajectoire de consigne admissible par le véhicule et navigable, c'est à dire sans obstacles, la trajectoire désirée $\mathcal{U}$ est calculée directement à partir des poses du graphe d'apprentissage, et peut être exprimée par un graphe de vecteurs de consigne tel que :

$$\mathcal{U} = \{\mathbf{u}_1^*, \mathbf{u}_2^*, \ldots, \mathbf{u}_n^*\}, \tag{8.2}$$

où chaque vecteur de consigne contient 5 valeurs :

$$\mathbf{u}^* = \{x^*, y^*, \psi^*, U^*, \dot{\psi}^*\}. \tag{8.3}$$

Le point de coordonnées $\mathbf{o}^* = \{x^*, y^*\}$ indique la position du point sur la trajectoire exprimée dans le repère de la base donnée, ψ^* est le cap du véhicule, U^* est la vitesse longitudinale désirée et $\dot{\psi}^*$ la vitesse angulaire désirée.

8.2.3 Loi de commande

Le problème de suivi de trajectoire peut être formulé comme le suivi d'un véhicule tel que défini dans Benhimane *et al.* (2005); Benhimane (2006). Dans le cas présent, le véhicule à suivre est un véhicule virtuel se déplaçant sur la trajectoire de consigne à une distance d de la position du véhicule courant.

Soit la pose courante du véhicule $\mathbf{T}_c$ estimée à l'instant t par la localisation visuelle et ramenée en 2 dimensions à une valeur de cap ψ et une position $\mathbf{o} = \begin{bmatrix} x & y \end{bmatrix}$. Il est possible d'extraire un vecteur de consigne $\mathbf{u}^*$ en projetant orthogonalement la pose courante sur le segment de la trajectoire $\mathcal{U}$ (un segment relie deux vecteurs de consignes successifs). Le vecteur désiré est alors interpolé à une distance d positionnée vers "l'avant" sur la trajectoire de consigne (*cf.* figure 8.4), afin de générer une erreur de pose pouvant être régulée par la loi de commande.

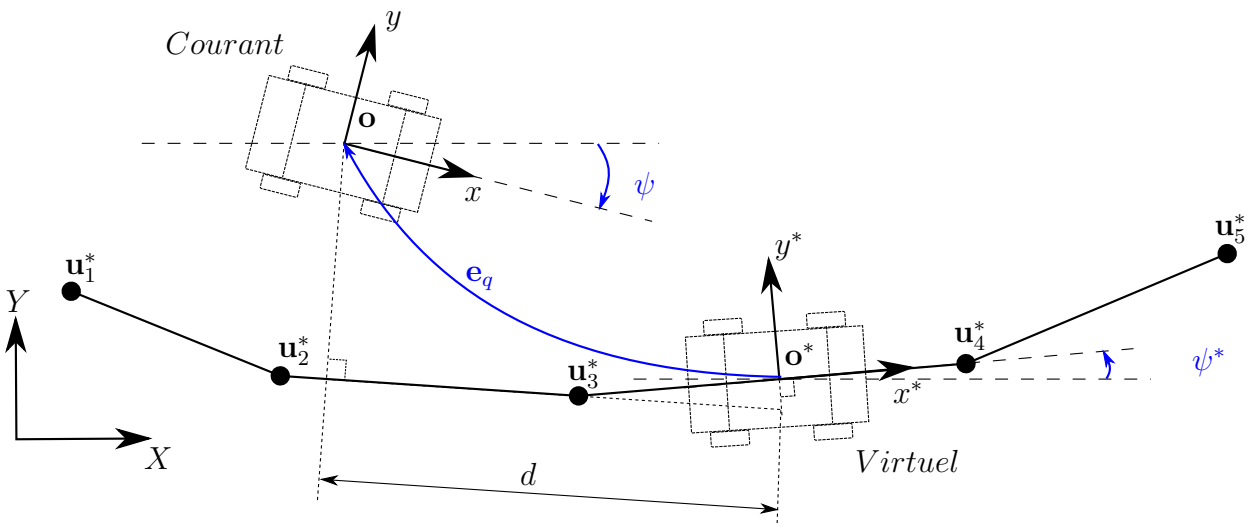


FIG. 8.4 – Erreur régulée lors du suivi de trajectoire. La position courante du véhicule est projetée sur le segment de la trajectoire le plus proche. La pose de référence est alors sélectionnée à une distance d en avant sur la trajectoire. Cela permet de définir les erreurs $\{\mathbf{e}_q, e_\psi\}$ entre le véhicule virtuel et le véhicule courant.

L'erreur en translation entre le point de référence et le point courant est définie par :

$$\mathbf{e}_q = \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \mathbf{R}_{\psi^*}^T(\mathbf{o} - \mathbf{o}^*) = \mathbf{R}_{\psi^*}^T \begin{bmatrix} x - x^* \\ y - y^* \end{bmatrix}, \tag{8.4}$$

où la matrice de rotation d'angle ψ^* s'écrit :

$$\mathbf{R}_{\psi^*} \begin{bmatrix} \cos(\psi^*) & -\sin(\psi^*) \\ \sin(\psi^*) & \cos(\psi^*) \end{bmatrix}. \tag{8.5}$$

L'erreur angulaire est quant à elle définie directement par :

$$e_\psi = \psi - \psi^* \tag{8.6}$$

Ces erreurs permettent de contrôler la vitesse longitudinale U et l'angle de braquage δ des roues du véhicule. La vitesse longitudinale peut être contrôlée à l'aide d'un retour d'état proportionnel à l'erreur longitudinale et la vitesse désirée. L'angle de braquage des roues δ dépend quant à lui de l'erreur transversale et de l'erreur en orientation. La loi de commande, dérivée de Benhimane *et al.* (2005) sans le retour d'état sur les vitesses, est la suivante :

$$\begin{cases} U = U^* - k_x(|U^*| + \epsilon)e_x \\ \dot{\psi} = \dot{\psi}^* - k_y|U^*|e_y - k_\psi|U^*|\tan(e_\psi) \end{cases}, \tag{8.7}$$

où les gains k_x, k_y,k_ψ et ϵ sont des scalaires positifs. La consigne sur l'angle de braquage est obtenue suivant l'équation (8.1) :

$$\delta = L \arctan\left(\frac{\dot{\psi}^*}{U^*}\right) \tag{8.8}$$

8.3 Détection d'obstacles

Afin de naviguer en toute sécurité, le laser situé à l'avant du véhicule est utilisé pour détecter les obstacles. La vitesse longitudinale U obtenue par la loi de commande est alors modulée proportionnellement à la distance minimale détectée dans la trace laser :

$$U_L = \begin{cases} 0 & \text{si} \quad d_{min} < d_{arret} \\ U\frac{d_{min}-d_{arret}}{d_{securite}-d_{arret}} & \text{sinon si} \quad d_{min} < d_{securite} \\ U & \text{sinon} \end{cases} \tag{8.9}$$

où $d_{securite}$ est la distance maximale de prise en compte des obstacles, d_{arret} est la distance d'arrêt total du véhicule et d_{min} est la distance minimale mesurée dans la trace laser. Lors des expérimentations, une distance de sécurité de 2 mètres et une distance d'arrêt de 1 mètre ont été utilisées, ce qui permet de définir une zone de décélération de 1 mètre, suffisante pour la vitesse du Cycab (1.2 m/s).

8.4 Station déportée

Pour permettre la visualisation des résultats à distance, les 6 degrés de liberté de la pose estimée du véhicule sont envoyés à la station déportée, par liaison WIFI, par protocole TCP. Cette station dispose d'une copie de la base de donnée embarquée sur le véhicule et permet à la fois d'afficher la trajectoire du véhicule et de générer une vue de synthèse correspondant à la position estimée, en fonction des images de référence de la base de donnée (*cf.* section 4.4.2). L'image envoyée par la caméra IP est également affichée à l'écran de la station.

8.5 Résultats expérimentaux

8.5.1 Sophia Antipolis

Les résultats suivants ont été obtenus à l'INRIA Sophia Antipolis sur une trajectoire d'apprentissage d'environ 100 mètres contenant deux virages à 90 degrés. La base de données reconstruite a été utilisée pour la localisation en ligne du véhicule et pour générer une trajectoire de consigne avec une vitesse linéaire constante de 1.2 mètres par seconde. La figure 8.5 montre la trajectoire de référence utilisée et la trajectoire courante estimée lors d'une expérience de navigation autonome, ainsi que quelques couples de poses clés. Les erreurs longitudinales et transversales correspondantes sont affichées sur la figure 8.6. L'erreur longitudinale a été centré en zéro par la valeur $d = 0.75m$, correspondant à la distance entre le véhicule virtuel et le véhicule courant désirée. La précision et la fréquence de la localisation visuelle ont permis de suivre la trajectoire désirée avec des erreurs transversales inférieures à 25 cm.

8.5.2 Clermont Ferrand

Le système complet de cartographie et de navigation autonome a également été testé et validé dans le cadre des démonstrations finales du projet CityVIP se déroulant au centre ville de Clermont-Ferrand (place de Jaude). Un apprentissage a été effectué sur une trajectoire d'environ 500 mètres. La base de données obtenue a été utilisée pour la localisation en ligne du véhicule et pour générer une trajectoire de consigne avec une vitesse linéaire constante de 1.2 mètres par seconde. La figure 8.7 montre une trajectoire de consigne ainsi

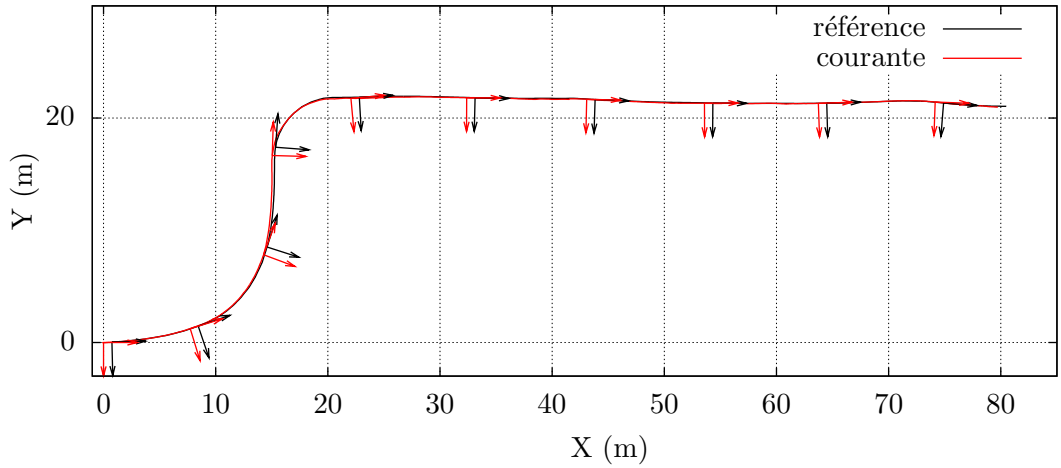


FIG. 8.5 – Trajectoire suivie à Sophia Antipolis. Le véhicule a suivi de manière automatique une trajectoire de 100 mètres. Quelques couples de poses véhicule virtuel/courant sont affichées. Les erreurs longitudinales et transversales correspondantes sont fournies sur la figure 8.6.

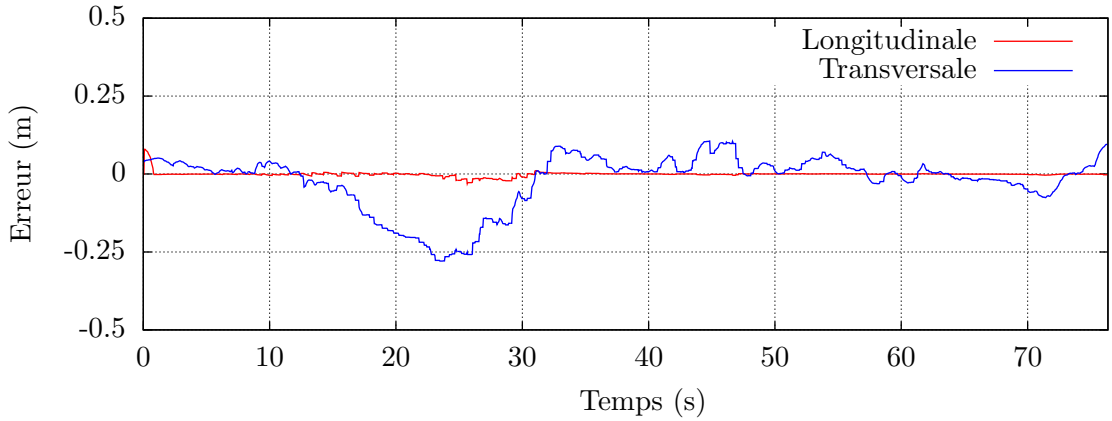


FIG. 8.6 – Erreurs de pose à Sophia Antipolis. L'erreur longitudinale à été centrée en zéro par la valeur de la distance d entre le véhicule virtuel et la position courante.

qu'une trajectoire estimée lors d'une expérience de navigation autonome. Le véhicule a suivi fidèlement la trajectoire désirée, tout le long de la séquence.

La figure 8.8 montre quelques photographies prises lors des expériences. Sur les images 8.8(a),8.8(b), le véhicule navigue sur une voie piétonne. Un nombre important de piétons occultent l'image perçue par le véhicule, et un grand nombre d'éléments de la scène ont changé par rapport à l'apprentissage (*i.e.* chaises de la terrasse du café). Cependant grâce aux techniques robustes utilisées dans la localisation visuelle, une pose précise est fournie à la loi de commande et permet une navigation fluide, robuste aux aberrations.

L'image 8.8(e) montre l'importance du module de détection d'obstacles, et ce même lors d'expérimentations. En effet, un piéton est venu intentionnellement se placer devant le véhicule, qui grâce au laser, a ralenti sa vitesse jusqu'à s'arrêter devant le piéton puis est reparti lorsque la voie s'est dégagée.

Les images 8.8(d),8.8(f),8.8(g) et 8.8(h), montrent d'autres portions de la trajectoire se déroulant sur la place. Ce type d'environnement a mis en évidence certaines limitations des algorithmes de vision. En effet la partie saillante de l'information se trouve sur les

façades des bâtiments (forts gradients photométriques), qui sont situés loin de la caméra (*cf.* figure 8.9). Bien que l'algorithme de sélection de pixels (*cf.* section 5.3) prenne en compte les gradients géométriques, les points du sol (proches de la caméra) sont très peu texturés. De plus ces pixels sont souvent mal reconstruits par la mise en correspondance dense (motifs répétés) et se situent sur une région de l'image sensible aux occultations (bas de l'image).

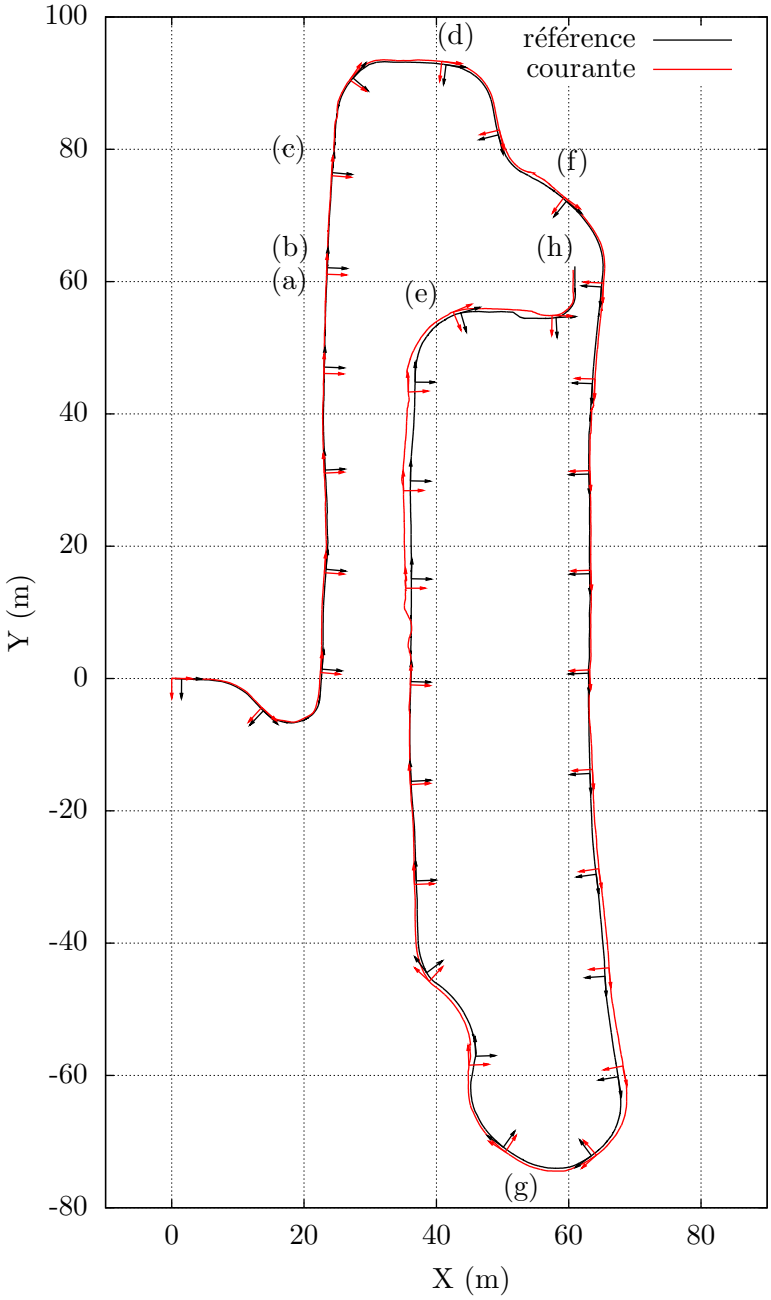


FIG. 8.7 – Trajectoire suivie à Clermont-Ferrand. Le véhicule a suivi de manière automatique une trajectoire de 490 mètres en environnement urbain. Les images correspondantes sont fournies en figure 8.8.

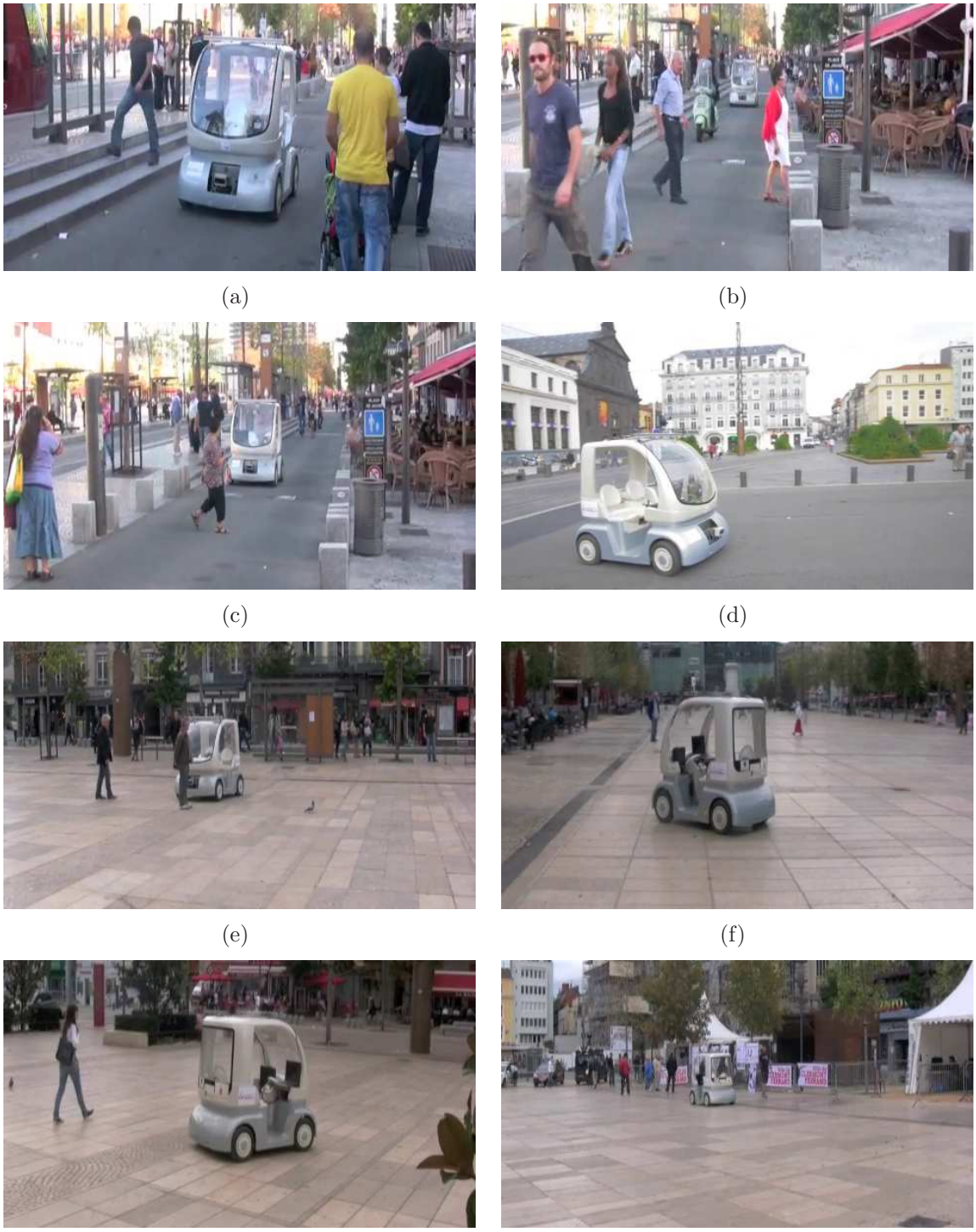


FIG. 8.8 – Quelques images capturées lors d'expériences de navigation autonome à Clermont-Ferrand. (a),(b),(c), des piétons. (d),(f),(g),(h) navigation sur la place.(e), détection d'obstacle.

Les caméras utilisées pour la localisation ont un très grand angle (distance focale de 1.6 mm), ce qui accentue l'effet d'invariance des façades aux translations, c'est à dire qu'un petit mouvement en translation de la caméra ne génère pas de changement dans l'image.

Fig. 8.9 – Exemple de façades à "l'infini".

(a) 10% des meilleurs pixels de l'image de reference

(b) Image courante vue par la caméra.

Fig. 8.10 – Exemple d'ombres portées. Les deux images ont étés capturées à une heure d'intervalle. Les ombres portées sont sélectionnées comme pixels saillants, mais se sont déplacées dans l'image courante.

Dans certains cas, l'estimation des translations peut être imprécise, ce qui peut perturber la navigation. Pour résoudre ce problème, il serait possible d'utiliser à la fois une caméra grand angle pour la robustesse et une caméra à longue focale pour la précision.

Un second type de scénario a mis en évidence un point améliorable de la technique de localisation visuelle. Sur la figure 8.10(a), les 10% des meilleurs pixels de l'image de référence sont affichés en rouge. On voit bien que les gradients des ombres portées des batîments et de la végétation sur la façade sont sélectionnés par l'algorithme de sélection de pixels. L'image 8.10(b) montre une image capturée 1 heure après l'image de référence, les ombres ont changé de position. Ces ombres génèrent de faux gradients dans la fonction de coût qui perturbent la localisation. Pour rendre la localisation plus robuste, il serait possible d'effectuer une sélection de pixels en ligne, afin de remettre en cause la sélection effectuée hors-ligne, qui n'est pas valable dans ce cas. Une seconde option serait d'améliorer la phase de sélection hors-ligne en détectant les ombres dans les images.

8.6 Conclusion

Dans ce chapitre, les algorithmes de cartographie et de localisation en ligne ont été utilisés dans le cadre d'une application de navigation autonome. Un véhicule de type Cycab a été contrôlé automatiquement en conditions réelles dans un environnement urbain. Pour effectuer la localisation 3D du véhicule, uniquement une caméra monoculaire a été utilisée, sans l'aide d'aucun autre capteur. Un laser est utilisé seulement pour la sécurité (détection des obstacles). La robustesse des algorithmes a pu être mise à l'épreuve sur des séquences contenant d'importantes occultations (*i.e.* piétons et véhicules). Pour simplifier la mise en œuvre et générer des trajectoires admissibles par le véhicule, la trajectoire de consigne utilisée est la même que celle utilisée lors de la phase d'apprentissage. Cependant, puisque l'algorithme fournit une localisation 3D, il est tout à fait possible de suivre automatiquement des trajectoires localement différentes de l'apprentissage (*i.e.* évitement d'obstacles, dépassement), tel qu'il a été montré dans les résultats du chapitre de localisation temps-réel (*cf.* section 6.2).

Pour améliorer la robustesse et la précision de la localisation visuelle, il serait possible d'utiliser simultanément deux caméras, l'une grand angle (robustesse) et l'autre à longue focale (précision). Ou utiliser deux caméras grand angle, l'une à l'avant du véhicule et l'autre à l'arrière, afin de maximiser la couverture visuelle et ainsi minimiser la quantité d'occultations et maximiser l'observabilité des mouvements 3D (*cf.* Baker *et al.* (2001)).

Quatrième partie

Conclusion et perspectives

Conclusions

Dans ces travaux, une nouvelle représentation sphérique ego-centrée de l'environnement a été présentée. Ce modèle est constitué d'un graphe d'images sphériques denses augmentées par la profondeur. Chaque nœud du graphe contient une sphère augmentée disponible en plusieurs résolutions. Les nœuds du graphe sont connectés par une pose 3D précise. Les données denses contenues dans les sphères permettent d'utiliser des techniques de transfert d'images, rendant la représentation parfaitement adaptée aux techniques de localisation visuelle directes. Pour construire ce modèle, un nouveau capteur d'images sphériques a été proposé, ainsi qu'une technique d'odométrie visuelle 3D directe, permettant un placement précis des nœuds du graphe dans l'espace. La méthode proposée permet de modéliser automatiquement, avec une représentation *dense*, de larges environnements urbains, tout en maintenant un maximum de précision locale.

Le modèle obtenu lors de cette phase d'apprentissage est utilisé pour localiser en temps réel une caméra naviguant dans le voisinage du graphe. Pour accomplir cette tâche, une nouvelle méthode de sélection de pixels, adaptée aux méthodes directes, a été développée. Cette approche permet d'accélérer les temps de calculs de l'alignement d'images en utilisant un nombre réduit de pixels, tout en maintenant l'observabilité complète de tous les mouvements 3D. L'avantage de cette approche est que la sélection des pixels est effectuée hors-ligne lors de la construction de la base de donnée. Lors de la localisation en ligne, seulement le nombre de pixels désiré est utilisé, ce qui permet d'accélérer le temps de calcul et d'obtenir une localisation temps-réel.

Pour améliorer la robustesse de la méthode de localisation visuelle en ligne, une approche hybride a été proposée. En effet les images de la base de données peuvent avoir étés reconstruites dans des conditions d'illumination différentes de celle perçues en ligne. L'approche proposée minimise simultanément une erreur basée modèle et une erreur basée odométrie visuelle, ce qui permet d'améliorer la robustesse aux changements d'illumination, mais aussi aux occultations.

La chaîne complète de modélisation automatique et re-localisation temps-réel a été intégrée dans un système de navigation autonome, et utilisée pour du contrôle automatique en environnements urbains. Grâce aux méthodes robustes utilisées, le véhicule est capable de naviguer dans des environnements réels, contenants piétons et autres véhicules, en utilisant seulement une caméra monoculaire pour la localisation.

Tous les algorithmes proposés dans cette thèse, sont le plus générique possible, et ne nécessitent aucun a priori sur l'environnement. La sélection d'information sur la sphère est la même que la sélection d'information sur un capteur perspectif, ou que pour le suivi d'une surface planaire. L'algorithme hybride robuste aux changements d'illuminations est également applicable à tous les capteurs et types de suivis directs. En effet, comme on peut le voir dans les résultats, la méthode de localisation a été utilisée en intérieur pour le positionnement d'une caméra déplacée manuellement. En extérieur, les algorithmes fonctionnent en environnement urbains fortement structuré, mais sont aussi efficaces en milieu moins structuré contenant beaucoup de végétation (*e.g.* INRIA Sophia Antipolis).

Perspectives

Les méthodes de cartographie et re-localisation ont été validées sur un certain nombre d'expérimentations, qui ont permis d'identifier plusieurs éléments améliorables.

Sur la construction du modèle d'apprentissage, l'usage de caméras stéréo en configuration divergente nécessite de redéfinir certains outils comme la mise en correspondance dense, qui pourrait être améliorée en prenant en compte par exemple les différences de résolution entre les images.

Comme il a été évoqué dans les conclusions, un des objectifs a été de développer des méthodes basées uniquement sur une information "capteur". Cependant, pour construire des bases de données valables à long terme, il semble indispensable d'inclure l'information sémantique dans la sélection de pixels. Par exemple, en zone urbaine, la végétation contient en général de forts gradients photométriques dans les images et est souvent sélectionnée comme information saillante. Cette information n'est pas du tout stable, d'une part en cas de vent, la végétation ne respecte pas l'hypothèse de corps rigide, et son aspect change au cours des saisons. Un second exemple peut être les ombres portées, qui génèrent aussi de forts gradients photométriques, et qui changent de position en quelques minutes en fonction du soleil.

Ensuite un aspect qui n'a pas été abordé dans la sélection d'information est l'incertitude sur la géométrie. En effet lors des expérimentations, une meilleure précision a été constatée lorsqu'un nombre réduit des pixels saillants est utilisé. Une hypothèse plausible serait que les pixels à fort gradient photométrique sont bien mis en correspondance par les méthodes de mise en correspondance dense, ce qui permet d'avoir une maximum de précision géométrique sur ces régions, donc une bonne localisation. Il serait intéressant d'ajouter à la sélection d'information l'incertitude géométrique sur les pixels.

Pour améliorer la robustesse de la méthode de localisation en ligne, il serait intéressant de sélectionner les meilleurs pixels en ligne, afin de rejeter certaines aberrations liées aux occultations ou aux changements d'apparence. L'utilisation d'une paire stéréo en ligne serait aussi un plus, pour par exemple détecter les obstacles, distinguer les changements d'illumination locaux des occultations et aussi détecter les changements géométriques dans la scène. Pour une exploitation à long terme des bases de données, il semble indispensable de mettre à jour les images de la base de donnée lors de la localisation, par exemple avec une méthode de *SLAM* temps-réel.

Enfin, les résultats ont montré que cette représentation ego-centrée permet la localisation localement autour des sphères. Il serait intéressant de quantifier cette région de validité, pour pouvoir l'agrandir, par exemple en prenant en compte dans les fonctions de *warping*, les changements de résolution dans les images, liés aux changements de point vue.

Cinquième partie

Annexes

Calcul des matrices Jacobiennes

Jacobienne courante

Chaque ligne de la matrice Jacobienne courante correspond à la dérivé la fonction (2.26) à l'état courant de la minimisation, c'est à dire en $\mathbf{x} = \mathbf{0}$:

$$\mathbf{J}(\mathbf{0}) = \left[\nabla_{\mathbf{x}} \boldsymbol{\mathcal{I}}(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z, \mathbf{p}^*))\right]_{\mathbf{x}=\mathbf{0}}. \tag{10}$$

En prenant en compte la propriété (2.15), on obtient :

$$\mathbf{J}(\mathbf{0}) = \left[\nabla_{\mathbf{x}} \boldsymbol{\mathcal{I}}(w(w(\mathbf{T}(\mathbf{x}), Z, \mathbf{p}^*)), \widehat{\mathbf{T}})\right]_{\mathbf{x}=\mathbf{0}}, \tag{11}$$

qui peut être écrit comme le produit de trois matrices Jacobiennes :

$$\mathbf{J}(\mathbf{0}) = \mathbf{J}_{\boldsymbol{\mathcal{I}}^w} \mathbf{J}_w \mathbf{J}_{\mathbf{T}} \tag{12}$$

1. $\mathbf{J}_{\boldsymbol{\mathcal{I}}^w}$ est de dimension 1×2 (la troisième composante étant égale à 0) et correspond à la dérivée spatiale des intensités des pixels de l'image de courante, transformées par la fonction de *warping* $w(\widehat{\mathbf{T}}; Z, \mathbf{p}^w)$:

$$\mathbf{J}_{\boldsymbol{\mathcal{I}}^w} = \left[\nabla_{\mathbf{p}^w} \boldsymbol{\mathcal{I}}(w(\widehat{\mathbf{T}}; Z, \mathbf{p}^w))\right]_{\mathbf{p}^w=\mathbf{p}^*}. \tag{13}$$

2. $\mathbf{J}_w$ est de dimension 3×12 et correspond à dérivé de la position des pixels par rapport à la fonction de *warping* (projection perspective, sphérique *etc*) :

$$\mathbf{J}_w = [\nabla_{\mathcal{Z}} w(\mathcal{Z}; Z, \mathbf{p}^*)]_{\mathcal{Z}=\mathbf{T}(\mathbf{0})=\mathbf{I}} \tag{14}$$

3. $\mathbf{J}_{\mathbf{T}}$ est de dimension 12×6 et correspond à la dérivée $\mathbf{T}(\mathbf{x})$ par $\mathbf{x}$:

$$\mathbf{J}_{\mathbf{T}} = [\nabla_{\mathbf{x}} \mathbf{T}(\mathbf{x})]_{\mathbf{x}=\mathbf{0}}. \tag{15}$$

   Cette matrice peut être formulée par :

$$\mathbf{J}_{\mathbf{T}} = \begin{bmatrix} \mathbf{a_1} & \mathbf{a_2} & \dots & \mathbf{a_6} \end{bmatrix}^T, \tag{16}$$

   où les vecteurs $\mathbf{a_i}$ sont obtenus en ré-arrangeant les matrices génératrices $\mathbf{A_i}$ sous forme vectorielle. Ces matrices génératrices sont des bases de l'algèbre de Lie $\mathfrak{se}(3)$ et peuvent être déterminées à partir la matrice de l'équation (2.7) en sélectionnant une base $\mathbf{x_i} = \boldsymbol{v}_i, \boldsymbol{\omega}_i$ (*i.e.* $\mathbf{x_1} = (1, 0, 0, 0, 0, 0)$) :

$$\mathbf{A_i} = \begin{bmatrix} [\boldsymbol{\omega}_i]_\times & \boldsymbol{v}_i \\ \mathbf{0} & 0 \end{bmatrix} \tag{17}$$

Jacobienne de référence

Chaque ligne de La matrice Jacobienne de référence $\mathbf{J}(\tilde{\mathbf{x}})$, peut être obtenue en dérivant l'équation (2.26) par :

$$\mathbf{J}(\mathbf{x}) = \left[\nabla_{\mathbf{x}}\boldsymbol{\mathcal{I}}(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z, \mathbf{p}^*))\right]_{\mathbf{x}=\tilde{\mathbf{x}}} \tag{18}$$

La propriété de groupe de l'équation (2.15), permettent de remplacer l'image courante par l'image de référence en introduisant la vraie transformation $\overline{\mathbf{T}}$ correspondant à la solution, tel que :

$$\mathbf{J}(\mathbf{x}) = \left[\nabla_{\mathbf{x}}\boldsymbol{\mathcal{I}}(w(w(\overline{\mathbf{T}}^{-1}\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z, \mathbf{p}^*)), \overline{\mathbf{T}})\right]_{\mathbf{x}=\tilde{\mathbf{x}}} = \left[\nabla_{\mathbf{x}}\boldsymbol{\mathcal{I}}^*(w(\overline{\mathbf{T}}^{-1}\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), Z, \mathbf{p}^*))\right]_{\mathbf{x}=\tilde{\mathbf{x}}} \tag{19}$$

Cette matrice Jacobienne peut être décomposée en un produit de trois matrices Jacobiennes :

$$\mathbf{J}(\tilde{\mathbf{x}}) = \mathbf{J}_{\boldsymbol{\mathcal{I}}^*}\mathbf{J}_{w^*}\mathbf{J}_{\mathbf{T}^*} \tag{20}$$

1. $\mathbf{J}_{\boldsymbol{\mathcal{I}}^*}$ est de dimension 1×2 (la troisième composante étant égale à 0) et correspond à la dérivée spatiale des intensités des pixels de l'image de référence, transformées par la fonction de *warping* $w(\overline{\mathbf{T}}^{-1}\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); Z, \mathbf{p}^w)$:

$$\mathbf{J}_{\boldsymbol{\mathcal{I}}^*} = \left[\nabla_{\mathbf{p}^w}\boldsymbol{\mathcal{I}}(w(\mathbf{I}; Z, \mathbf{p}^w))\right]_{\mathbf{p}^w=\mathbf{p}^*} \tag{21}$$

   où la transformation $\overline{\mathbf{T}}^{-1}\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x})) = \mathbf{I}$ en $\mathbf{x} = \tilde{\mathbf{x}}$.

2. $\mathbf{J}_{w^*}$ est de dimensions 2×12 et correspond à dériver la position des pixels par rapport à la fonction de *warping* (projection perspective, sphérique *etc*) :

$$\mathbf{J}_{w^*} = [\nabla_{\mathcal{Z}} w(\mathcal{Z}; Z, \mathbf{p}^*)]_{\mathcal{Z}=\mathbf{I}} = \mathbf{J}_w \tag{22}$$

3. $\mathbf{J}_{\mathbf{T}^*}$ dépend du déplacement inconnu $\tilde{\mathbf{x}}$, qui est la solution du problème d'estimation. Cette dérivée peut être formulée sous la forme :

$$\left[\frac{\partial(\overline{\mathbf{T}}^{-1}\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}))}{\partial\mathbf{x}}\right]_{\mathbf{x}=\tilde{\mathbf{x}}}\tilde{\mathbf{x}} = \left[\frac{\partial(\mathbf{T}(\mathbf{x}))}{\partial\mathbf{x}}\right]_{\mathbf{x}=\mathbf{0}}\tilde{\mathbf{x}}. \tag{23}$$

   En considérant le partie gauche de l'équation, il est possible de substituer la variable $\mathbf{x}$ par $\tilde{\mathbf{x}} + \mathbf{y}$:

$$\mathbf{J}_{\mathbf{T}^*} = \left[\frac{\partial(\overline{\mathbf{T}}^{-1}\widehat{\mathbf{T}}\mathbf{T}(\tilde{\mathbf{x}} + \mathbf{y}))}{\partial\mathbf{x}}\right]_{\mathbf{x}=\mathbf{0}}\frac{\partial\mathbf{y}}{\partial\mathbf{x}}\tilde{\mathbf{x}} \tag{24}$$

   En appliquant les propriétés du groupe, on a $\mathbf{T}(\tilde{\mathbf{x}})\mathbf{T}(\mathbf{y}) = \mathbf{T}(\tilde{\mathbf{x}} + \mathbf{y})$, si l'on assume que la vraie pose $\overline{\mathbf{T}} \approx \widehat{\mathbf{T}}\mathbf{T}(\tilde{\mathbf{x}})$, on a :

$$\mathbf{J}_{\mathbf{T}^*} = \left[\frac{\partial\mathbf{T}(\mathbf{y})}{\partial\mathbf{y}}\right]_{\mathbf{y}=\mathbf{0}}\tilde{\mathbf{x}}, \tag{25}$$

   ce qui est équivalent à la partie droite de l'équation (23) pour tout $\mathbf{y} = \mathbf{x} - \tilde{\mathbf{x}}$. On a donc

$$\mathbf{J}_{\mathbf{T}^*}\mathbf{x} = \mathbf{J}_{\mathbf{T}}\mathbf{x}. \tag{26}$$

Récapitulatif

Pour la méthode de minimisation IC, la matrice Jacobienne de référence est utilisée :

$$\mathbf{J}_{ic} = \mathbf{J}(\tilde{\mathbf{x}}) = \mathbf{J}_{\boldsymbol{\mathcal{I}}^*}\mathbf{J}_w\mathbf{J}_{\mathbf{T}}, \tag{27}$$

tous les termes sont alors constants au cours de la minimisation.

Pour la méthode de minimisation ESM, les deux matrices Jacobiennes sont utilisées :

$$\mathbf{J}_{esm} = \frac{1}{2}(\mathbf{J}(\tilde{\mathbf{x}}) + \mathbf{J}(\mathbf{0})) = \frac{1}{2}(\mathbf{J}_{\boldsymbol{\mathcal{I}}^*} + \mathbf{J}_{\boldsymbol{\mathcal{I}}^w})\mathbf{J}_w\mathbf{J}_{\mathbf{T}}, \tag{28}$$

seulement la matrice $\mathbf{J}_{\boldsymbol{\mathcal{I}}^w}$ doit être mis à jour à chaque itération de la minimisation.

Gradients géométriques

La partie géométrique de la Jacobienne est définie par :

$$\mathbf{J}_G = \mathbf{J}_w\mathbf{J}_{\mathbf{T}} \tag{29}$$

La matrice $\mathbf{J}_{\mathbf{T}}$ est définie selon les bases de l'algèbre de Lie par :

$$\mathbf{J}_{\mathbf{T}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{30}$$

La matrice $\mathbf{J}_w$ peut être décomposée en deux matrices :

$$\mathbf{J}_w = \mathbf{J}_\Pi\mathbf{J}_R \tag{31}$$

où $\mathbf{J}_R$ de dimensions 3×12 est la dérivée d'une transformation rigide d'un point $\mathbf{P} = (X, Y, Z) \in \mathbb{R}^3$:

$$\mathbf{P}' = \mathbf{R}\mathbf{P} + \mathbf{t}, \tag{32}$$

par rapport aux 12 éléments de la transformation $\mathbf{T}$:

$$\mathbf{J}_R = \frac{\partial \mathbf{P}'}{\partial \mathbf{T}} = \begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & Y & Z & 1 \end{bmatrix} \tag{33}$$

Dans ce cas le produit $\mathbf{J}_{\mathbf{x}} = \mathbf{J}_R\mathbf{J}_{\mathbf{T}}$ donne :

$$\mathbf{J}_{\mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 & Z & -Y \\ 0 & 1 & 0 & -Z & 0 & X \\ 0 & 0 & 1 & Y & -X & 0 \end{bmatrix} \tag{34}$$

La matrice jacobienne de projection $\mathbf{J}_\Pi$ dépend du modèle de projection de l'image de référence.

– **Projection perspective**, $\Pi : \mathbb{R}^3 \to \mathbb{R}^2$:

$$\overline{\mathbf{p}^*} = \frac{\mathbf{K}\mathbf{P}'}{\mathbf{e}_3^T\mathbf{P}'}, \tag{35}$$

En $\mathbf{x} = \mathbf{0}$, $\mathbf{T}(\mathbf{0}) = \mathbf{I}$, on a alors $\mathbf{P}' = \mathbf{P}$:

$$\mathbf{J}_\Pi = \begin{bmatrix} f/Z & 0 & -(Xf)/Z^2 \\ 0 & fr/Z & -(Yf)/Z^2 \\ 0 & 0 & 0 \end{bmatrix} \tag{36}$$

La partie géométrique est alors égale à :

$$\mathbf{J}_G = \mathbf{J}_\Pi \mathbf{J}_\mathbf{x} = \begin{bmatrix} f/Z & 0 & -(Xf)/Z^2 & -(XYf)/Z^2 & f+(X^2f)/Z^2 & -(Yf)/Z \\ 0 & fr/Z & -(Yfr)/Z^2 & -fr-(Y^2fr)/Z^2 & (XYfr)/Z^2 & (Xfr)/Z \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{37}$$

En remplaçant les points 3D $\mathbf{P} = (X, Y, Z)$ par leur équivalent dans l'image de référence $(\mathbf{p}^* = (u, v)), Z)$ obtenus par $\mathbf{P} = Z\mathbf{K}^{-1}\mathbf{p}^*$ on obtient :

$$\mathbf{J}_G = \begin{bmatrix} f/Z & 0 & (u_0-u)/Z & -((u_0-u)(v_0-v))/fr & f+(u_0-u)^2/f & (f(v_0-v))/fr \\ 0 & fr/Z & (v_0-v)/Z & -fr-(v_0-v)^2/fr & ((u_0-u)(v_0-v))/f & -(fr(u_0-u))/f \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{38}$$

où (f, fr, u_0, v_0) sont les paramètres intrinsèques de la matrice $\mathbf{K}$ de l'image de référence (le facteur de cisaillement $s = 0$). On peut voir que seulement les trois premières colonnes (translations) dépendent de la profondeur Z.

– **Projection sphérique** : $\Pi : \mathbb{R}^3 \to \mathbb{R}^2$:

$$\mathbf{q}_E = \frac{\mathbf{P}'}{\|\mathbf{P}'\|}. \tag{39}$$

En $\mathbf{x} = \mathbf{0}$, $\mathbf{T}(\mathbf{0}) = \mathbf{I}$, on a alors $\mathbf{P}' = \mathbf{P}$, les points $\mathbf{q}_E$ en coordonnées cartésiennes peuvent être convertis en coordonnées sphériques par :

$$\begin{bmatrix} \theta \\ \phi \\ \rho \end{bmatrix} = \begin{bmatrix} \arctan(Z/X) \\ \arctan(Y/\sqrt{X^2+Z^2}) \\ \sqrt{X^2+Y^2+Z^2} \end{bmatrix} \tag{40}$$

La matrice Jacobienne est donc définie telle que :

$$\mathbf{J}_\Pi = \begin{bmatrix} -Z/(X^2+Z^2) & 0 & X/(X^2+Z^2) \\ -(XY)/(\sqrt{X^2+Z^2}\rho^2) & \sqrt{X^2+Z^2}/\rho^2 & -(YZ)/(\sqrt{X^2+Z^2}\rho^2) \\ 0 & 0 & 0 \end{bmatrix} \tag{41}$$

La partie géométrique est alors égale à :

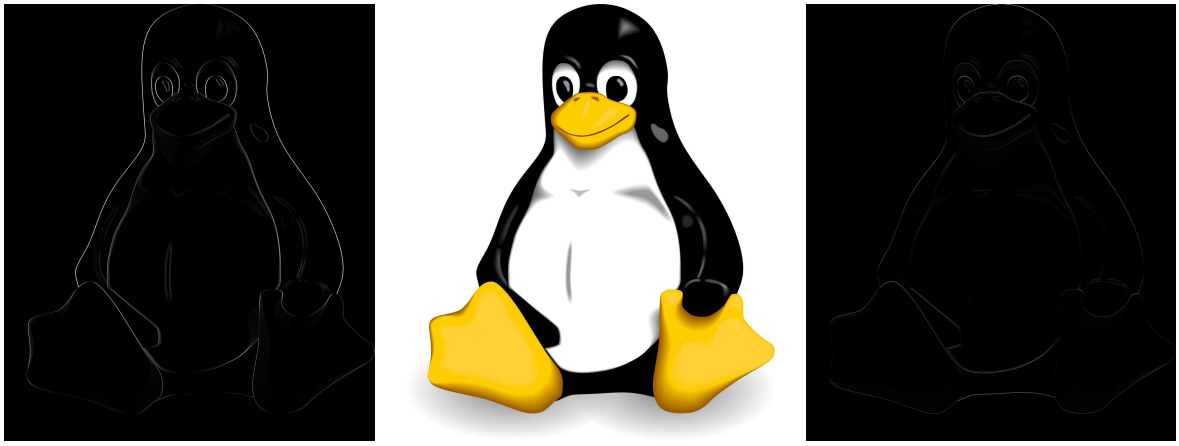
$$\mathbf{J}_G = \mathbf{J}_\Pi \mathbf{J}_\mathbf{x} = \begin{bmatrix} -Z/(X^2+Z^2) & 0 & X/(X^2+Z^2) & (XY)/(X^2+Z^2) & -1 & (YZ)/(X^2+Z^2) \\ -(XY)/(\sqrt{X^2+Z^2}\rho^2) & \sqrt{X^2+Z^2}/\rho^2 & -YZ/(\sqrt{X^2+Z^2}\rho^2) & -Z/\sqrt{X^2+Z^2} & 0 & X/\sqrt{X^2+Z^2} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{42}$$

Il faut noter la singularité en $(X = 0, Z = 0)$ de la dérivée de la projection sphérique. Cependant, ces points correspondent aux pôles de la sphère et ne sont jamais utilisés lors du suivi (*e.g.* ciel et sol).

Gradients photométriques

Les matrices $\mathbf{J}_\mathcal{I}$, correspondent à la dérivée spatiale des intensités des images par rapport aux pixels. Elles sont obtenus de manière numérique au pixel $\mathbf{p} = (u, v)$ par les opérateurs de gradient suivant les deux directions (*cf.* figure 11) :

$$\nabla_u = \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \nabla_v = \nabla_u^T = \frac{1}{2}\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{43}$$



(a) $\nabla_u \boldsymbol{\mathcal{I}}$ (b) $\boldsymbol{\mathcal{I}}$ (c) $\nabla_v \boldsymbol{\mathcal{I}}$

FIG. 11 – Gradients photométriques,

Les gradients correspondants dans les images sont alors obtenus par convolution :

$$\begin{aligned} \nabla_u \boldsymbol{\mathcal{I}}(u,v) &= \frac{\boldsymbol{\mathcal{I}}(u+du,v) - \boldsymbol{\mathcal{I}}(u-du,v)}{2du} \\ \nabla_v \boldsymbol{\mathcal{I}}(u,v) &= \frac{\boldsymbol{\mathcal{I}}(u,v+dv) - \boldsymbol{\mathcal{I}}(u,v-dv)}{2dv} \\ \nabla_z \boldsymbol{\mathcal{I}}(u,v) &= 0 \end{aligned} \tag{44}$$

Pour une image classique, on a $du = 1$ et $dv = 1$ car les coordonnées sont directement exprimées en pixels. Pour une image sphérique, les pas correspondent aux pas d'échantillonnage de la sphère $d\theta$ et $d\phi$ définis en section 3.2.5.

On a alors :

$$\mathbf{J}_{\boldsymbol{\mathcal{I}}} = \begin{bmatrix} \nabla_u \boldsymbol{\mathcal{I}}(u,v) \\ \nabla_v \boldsymbol{\mathcal{I}}(u,v) \\ 0 \end{bmatrix}^T \tag{45}$$

Il existe également d'autres opérateurs de gradients plus robustes au bruit, tels que les gradients de Sobel ($c = 2$) et de Prewitt ($c = 1$) :

$$\nabla_u = \frac{1}{a} \begin{bmatrix} -1 & 0 & 1 \\ -c & 0 & c \\ -1 & 0 & 1 \end{bmatrix}, \quad \nabla_v = \nabla_u^T = \frac{1}{a} \begin{bmatrix} -1 & -c & -1 \\ 0 & 0 & 0 \\ 1 & c & 1 \end{bmatrix}, \tag{46}$$

où $a = \sum |\nabla_u|$. Cependant, ces opérateurs lissent les contours ce qui réduit la précision lors de l'alignement des images.

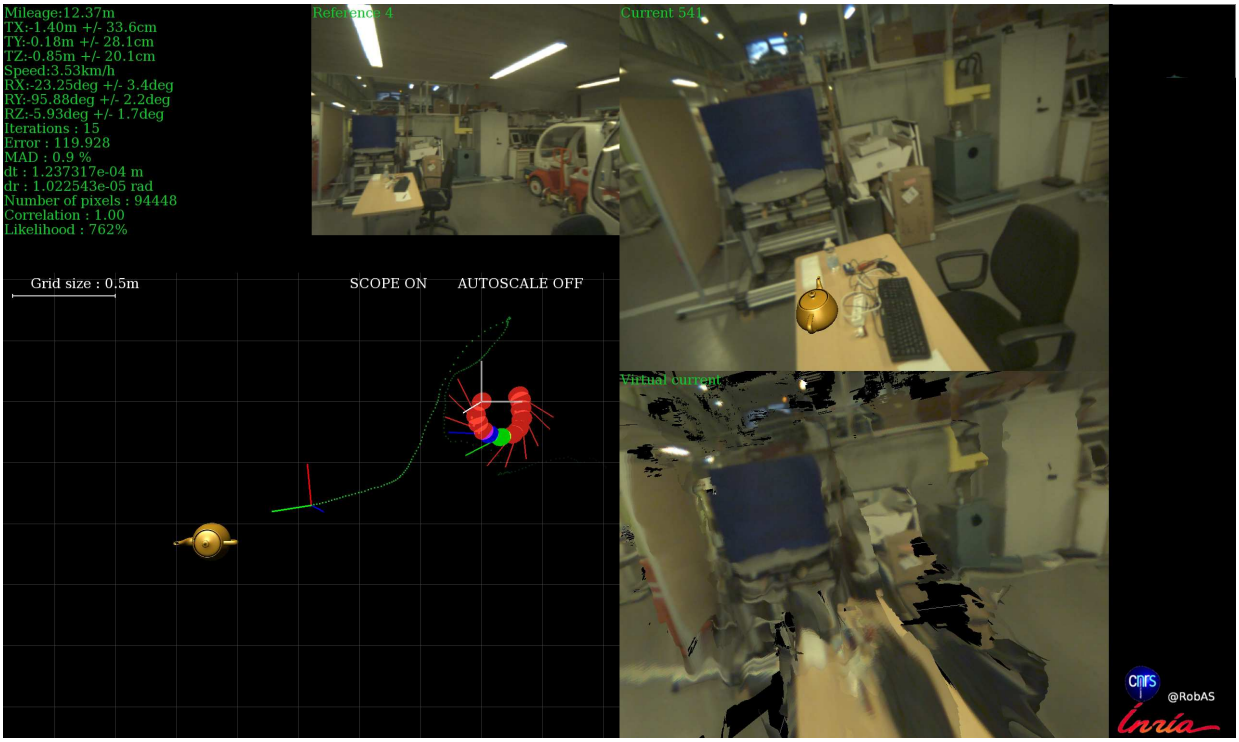


Fig. 12 – Interface DECSlam.

DECSlam

Un logiciel implémenté en C++ a été déposé comme application INRIA sous le nom de DECSlam, pour *Dense Ego-Centric Simultaneous Localisation and Mapping*. Il est composé de trois librairies indépendantes :

1. DECSlam_core : Contient toutes les fonctions de mathématiques et de traitement d'images nécessaire à la cartographie et au suivi visuel temps réel.
2. DECSlam_display : Contient une interface graphique réalisée avec SDL et optimisée pour OpenGL *cf.* figure 12.
3. DECSlam_capture : Contient les fonctions d'acquisition pour une ou plusieurs caméras firewire.

Des exécutables permettent de tester le système complet de cartographie et de localisation temps réel en utilisant une paire de caméras stéréo calibrée, en trois étapes comme défini sur la figure 13 :

1. stereo_grabber : Acquisition d'une séquence d'images stéréo.
2. stereo_offline_tracker : Construction d'une base de donnée ego-centrée d'images augmentées.
3. online_tracker : Permet d'effectuer une localisation temp-réel à partir d'une caméra monoculaire/stéréo, ou à partir d'une séquence d'images stockées sur disque dur en utilisant une base de données.

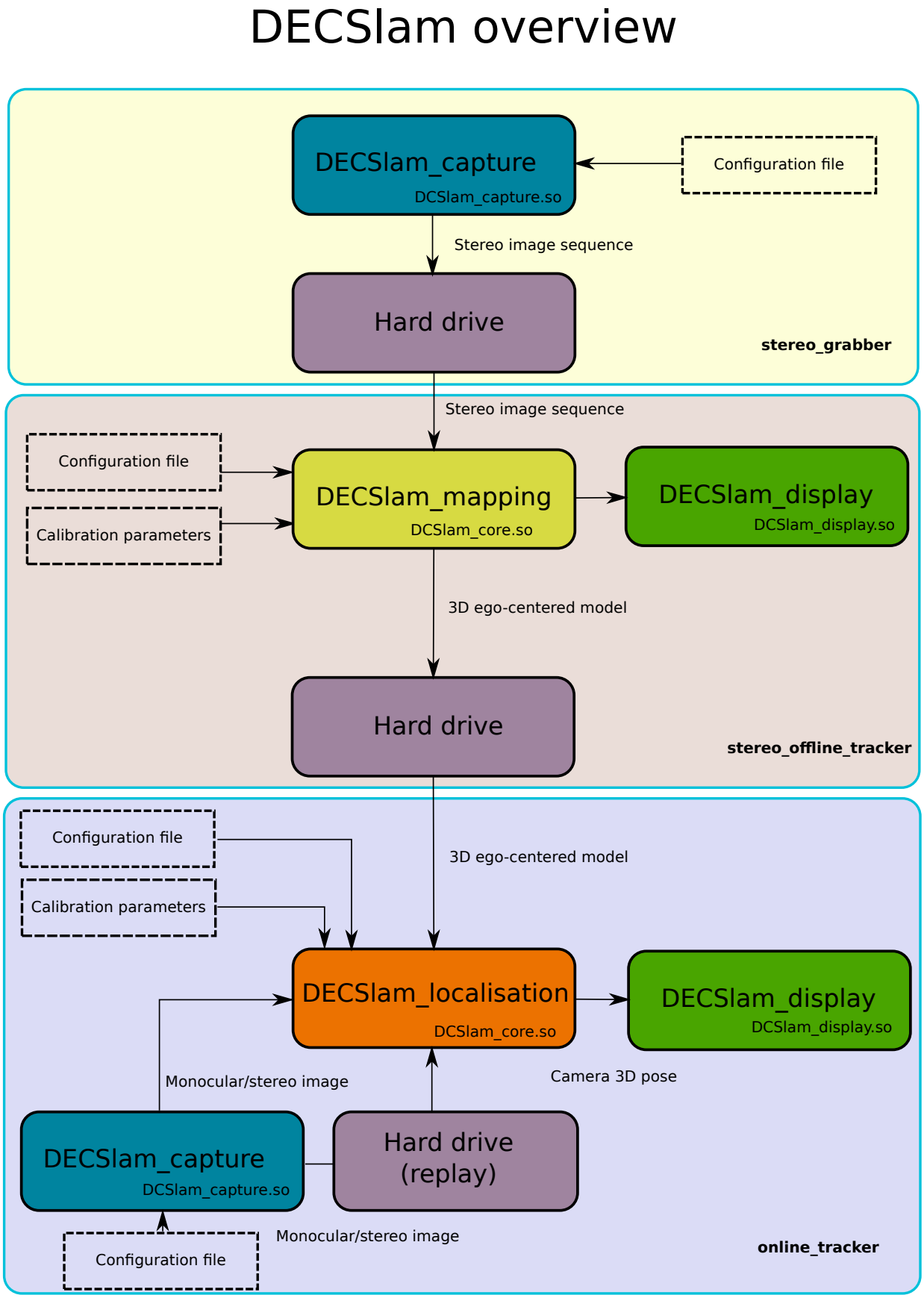


Fig. 13 – DECSlam.

Références

Audras, C., Comport, A.I., Meilland, M. & Rives, P. (2011). Real-time appearance-based slam for rgb-d sensors. In *Australian Conference on Robotics and Automation*. 7, 48

Avidan, S. & Shashua, A. (1997). Novel view synthesis in tensor space. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1034. 24

Baker, P., Fermuller, C., Aloimonos, Y. & Pless, R. (2001). A spherical eye from multiple cameras. *IEEE International Conference on Computer Vision and Pattern Recognition*, 576. 40, 46, 123

Baker, S. & Matthews, I. (2001). Equivalence and efficiency of image alignment algorithms. In *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 1090. 15, 29, 71, 72, 100

Bartoli, A. (2008). Groupwise geometric and photometric direct image registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**, 2098 –2108. 99, 101, 102

Basri, R. & Jacobs, D. (2001). Photometric stereo with general, unknown lighting. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 374–381. 100

Bay, H., Tuytelaars, T. & Gool, L.V. (2006). Surf : Speeded up robust features. In *European Conference on Computer Vision*, 404–417. 14, 70

Benhimane, S. (2006). *Vers une approche unifiee pour le suivi temps-reel et l'asservissement visuel*. Docteur en sciences - specialite : Informatique temps-reel, automatique et robotique, Ecole Nationale Superieure des Mines de Paris. 30, 71, 116

Benhimane, S. & Malis, E. (2004). Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE International Conference on Intelligent Robots and Systems*, 943–948. 30

Benhimane, S., Malis, E., Rives, P. & Azinheira, J. (2005). Vision-based control for car platooning using homography decomposition. In *IEEE International Conference on Robotics and Automation*, 2161 – 2166. 116, 117

Benhimane, S., Ladikos, A., Lepetit, V. & Navab, N. (2007). Linear and quadratic subsets for template-based tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–6. 70

Blinn, J.F. (1977). Models of light reflection for computer synthesized pictures. *SIGGRAPH Computer Graphics*, **11**, 192–198. 100

Bouguet, J.Y. (2005). Calibration toolbox. Http ://www.vision.caltech.edu/bouguetj/. 49, 51

Brown, D.C. (1971). Close-range camera calibration. *Photogrammetric engineering*, **37**, 855–866. 17

Burt, P.J. & Adelson, E.H. (1983). A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, **2**, 217–236. 32, 56

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**, 679–698. 70

Cappelle, C., El Najjar, M., Charpillet, F. & Pomorski, D. (2011). Virtual 3d city model for navigation in urban areas. *Journal of Intelligent and Robotic Systems*, 1–23. 17

Caron, G., Marchand, E. & Mouaddib, E. (2011). Tracking planes in omnidirectional stereovision. In *IEEE International Conference on Robotics and Automation*, 6306–6311. 15, 48

Caron, G., Dame, A. & Marchand, E. (2012). L'information mutuelle pour l'estimation visuelle directe de pose. In *RFIA, Reconnaissance des Formes et Intelligence Artificielle*. 18

Cascia, M.L. & Sclaroff, S. (1999). Fast, reliable head tracking under varying illumination. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 322–336. 100

Chapoulie, A., Rives, P. & Filliat, D. (2011). A spherical representation for efficient visual loop closing. In *Proceedings of the 11th workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*. 16, 65, 91

Cherubini, A. & Chaumette, F. (2011). Visual navigation with obstacle avoidance. In *IEEE International Conference on Intelligent Robots and Systems*, 1593–1598. 19

Cobzas, D., Zhang, H. & Jagersand, M. (2003). Image-based localization with depth-enhanced image map. In *IEEE International Conference on Intelligent Robots and Systems*, 1570–1575. 19, 39, 48

Comport, A.I. (2005). *Robust real-time 3D tracking of rigid and articulated objects for augmented reality and robotics*. Ph.D. thesis, Université de Rennes 1, Mention informatique. 17

Comport, A.I., Marchand, E., Pressigout, M. & Chaumette, F. (2006). Real-time markerless tracking for augmented reality : the virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics*, **12**, 615–628. 17

Comport, A.I., Malis, E. & Rives, P. (2007). Accurate quadrifocal tracking for robust 3d visual odometry. In *IEEE International Conference on Robotics and Automation*, 40–45. 15, 19, 71

Comport, A.I., Malis, E. & Rives, P. (2010). Real-time quadrifocal visual odometry. *The International Journal of Robotics Research*, **29**, 245–266. 16, 19, 21, 31, 102

Comport, A.I., Meilland, M. & Rives, P. (2011). An asymmetric real-time dense visual localisation and mapping system. In *First International Workshop on Live Dense Reconstruction from Moving Cameras In conjunction with the International Conference on Computer Vision*, Barcelona, Spain. 7

Cook, R.L. & Torrance, K.E. (1982). A reflectance model for computer graphics. *ACM Transactions on Graphics*, **1**, 7–24. 100

Courbon, J., Mezouar, Y. & Martinet, P. (2008). Indoor navigation of a non-holonomic mobile robot using a visual memory. *Autonomous Robots*, **25**, 253–266. 39

Courbon, J., Mezouar, Y. & Martinet, P. (2009). Autonomous navigation of vehicles from a visual memory using a generic camera model. *Intelligent Transport System*, **10**, 392–402. 19

Craciun, D., Paparoditis, N. & Schmitt, F. (2010). Multi-view scans alignment for 3d spherical mosaicing in large-scale unstructured environments. *Computer Vision and Image Understanding*, **114**, 1248 – 1263, special issue on Embedded Vision. 18

Cummins, M. & Newman, P. (2008). FAB-MAP : Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, **27**, 647–665. 16, 91

Dame, A. & Marchand, E. (2010). Accurate real-time tracking using mutual information. In *IEEE Symposium on Mixed and Augmented Reality*, 47–56. 15, 100

Dame, A. & Marchand, E. (2011). A new information theoretic approach for appearance-based visual path following. In *IEEE International Conference on Robotics and Automation*, 2459–2464, Shanghai, China. 19

Davison, A. & Murray, D. (2002). Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 865–880. 16

Debevec, P.E., Taylor, C.J. & Malik, J. (1996). Modeling and rendering architecture from photographs : a hybrid geometry- and image-based approach. In *Computer graphics and interactive techniques*, 11–20. 20, 66

Dellaert, F. & Collins, R. (1999). Fast image-based tracking by selective pixel integration. In *ICCV Workshop on Frame-Rate Vision*, 1–22. 70

Diosi, A. & Kleeman, L. (2007). Fast laser scan matching using polar coordinates. *The International Journal of Robotics Research*, **26**, 1125–1153. 14

Drummond, T., Society, I.C. & Cipolla, R. (2002). Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 932–946. 17

Duchon, C.E. (1979). Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology*, **18**, 1016–1022. 26

Duda, R. & Hart, P. (1971). Use of the hough transformation to detect lines and curves in pictures. Tech. Rep. 36. 70

Durrant-Whyte, H. & Bailey, T. (2006). Simultaneous localisation and mapping (slam) : Part i the essential algorithms. *IEEE Robotics and Automation Magazine*, **13**, 99–110. 16

Dyer, F.C. (1996). Spatial memory and navigation by honeybees on the scale of the foraging range. *The Journal of Experimental Biology*, **199**, 147–154. 5

Faugeras, O. (1993). *Three-dimensionnal computer vision : a geometric viewpoint*. MIT Press Cambridge, MA. 23

Fischler, M.A. & Bolles, R.C. (1981). Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *ACM Communications*, **24**, 381–395. 14

Furukawa, Y. & Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**, 1362–1376. 16

Fusiello, A., Trucco, E. & Verri, A. (2000). A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, **12**, 16–22. 52

Gallegos, G., Meilland, M., Rives, P. & Comport, A.I. (2010). Appearance-based slam relying on a hybrid laser/omnidirectional sensor. In *IEEE International Conference on Intelligent Robots and Systems*, 3005 –3010. 7, 17, 48

Geiger, A., Roser, M. & Urtasun, R. (2010). Efficient large-scale stereo matching. In *Asian Conference on Computer Vision*. 53, 55

Geyer, C. & Daniilidis, K. (2000). A unifying theory for central panoramic systems and practical applications. In *European Conference on Computer Vision*, 445–461. 43

Gonçalves, T. & Comport, A.I. (2011). Real-time direct tracking of color images in the presence of illumination variation. In *IEEE International Conference on Robotics and Automation*, 4417 –4422. 100, 102

Gorski, K.M., Hivon, E., Banday, A.J., Wandelt, B.D., Hansen, F.K., Reinecke, M. & Bartelman, M. (2005). Healpix – a framework for high resolution discretization, and fast analysis of data distributed on the sphere. *The Astrophysical Journal*, **622**, 759–773. 57, 58

Gortler, S.J., Grzeszczuk, R., Szeliski, R. & Cohen, M.F. (1996). The lumigraph. In *Computer graphics and interactive techniques*, 43–54. 20, 66

Grisetti, G., Grzonka, S., Stachniss, C., Pfaff, P. & Burgard, W. (2007). Efficient estimation of accurate maximum likelihood maps in 3d. In *IEEE International Conference on Intelligent Robots and Systems*, 3472 –3478. 16, 65

Grisetti, G., Stachniss, C., Grzonka, S. & Burgard, W. (2009). Toro - tree-based network optimizer. Http ://openslam.org/toro.html. 65

Guizzo, E. (2011). Google car. Http ://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works. 5

Hager, G. & Belhumeur, P. (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 1025 –1039. 31, 100

Hammoudi, K., Dornaika, F., Soheilian, B. & Paparoditis, N. (2010). Generating raw polygons of street facades from a 2d urban map and terrestrial laser range data. In *SSSI Australasian Remote Sensing and Photogrammetry Conference*. 18

Harris, C. & Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, 147–151. 14, 69

Hartley, R.I. & Sturm, P.F. (1995). Triangulation. In *International Conference on Computer Analysis of Images and Patterns*, 190–197. 55

Hartley, R.I. & Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edn. 23, 86

Hayet, J. (2003). *Contribution à la navigation d'un robot mobile sur amers visuels texturés dans un environnement structuré*. Ph.D. thesis, Université Paul Sabatier, Toulouse III. 70

Heikkila, J. & Silven, O. (1997). A four-step camera calibration procedure with implicit image correction. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 1106–. 23, 24

Henry, P., Krainin, M., Herbst, E., Ren, X. & Fox, D. (2010). Rgb-d mapping : Using depth cameras for dense 3d modeling of indoor environments. In *International Symposium on Experimental Robotics*. 48

Hirschmuller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**, 328–341. 46, 53, 55

Hirschmuller, H. & Scharstein, D. (2009). Evaluation of stereo matching costs on images with radiometric differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**, 1582–1599. 53

Howard, A. (2008). Real-time stereo visual odometry for autonomous ground vehicles. In *IEEE International Conference on Intelligent Robots and Systems*. 16

Huber, P. (1981). *Robust Statistics*. New york, Wiley. 32

Irani, M. & Anandan, P. (1998). Robust multi-sensor image alignment. In *International Conference on Computer Vision*, 959 –966. 15, 100

Irani, M. & Anandan, P. (2000). About direct methods. In *Vision Algorithms : Theory and Practice*, vol. 1883 of *Lecture Notes in Computer Science*, 267–277. 15

Irschara, A., Zach, C., Frahm, J.M. & Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2599 –2606. 18

Itti, L., Koch, C. & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 1254–1259. 70

Jazwinski, A.H. (1970). *Stochastic processes and filtering theory*. Acad. Press, 3rd edn. 16

Jogan, M. & Leonardis, A. (2000). Robust localization using panoramic view-based recognition. *IEEE International Conference on Computer Vision and Pattern Recognition*, 4136. 19, 39

Kadir, T. & Brady, M. (2001). Saliency, scale and image description. *International Journal of Computer Vision*, **45**, 83–105. 70

Ke, Y. & Sukthankar, R. (2004). Pca-sift : a more distinctive representation for local image descriptors. In *IEEE International Conference on Computer vision and pattern recognition*, 506–513. 70

Keys, R. (1981). Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **29**, 1153 – 1160. 26

Kim, H. & Hilton, A. (2009). Environment modelling using spherical stereo imaging. In *International Conference On 3-D Digital Imaging and Modeling*, 1534–1541. 48

Kitt, B., Geiger, A. & Lategahn, H. (2010). Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *IEEE Intelligent Vehicles Symposium*. 16

Klein, G. & Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 1–10. 16

Kolmogorov, V. & Zabih, R. (2001). Computing visual correspondence with occlusions using graph cuts. In *International Conference on Computer Vision*, vol. 2, 508–515. 53

Konolige, K. (2010). Sparse sparse bundle adjustment. In *British Machine Vision Conference*. 16

Konolige, K. & Agrawal, M. (2008). Frameslam : from bundle adjustment to realtime visual mapping. *IEEE Transactions on Robotics*, **24**, 1066–1077. 16

Krishnan, G. & Nayar, S. (2009). Towards A True Spherical Camera. In *SPIE Human Vision and Electronic Imaging*. 43

Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K. & Burgard, W. (2011). g2o : A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation*. 16, 68

Lafarge, F. & Mallet, C. (2011). Building large urban environments from unstructured point data. In *IEEE International Conference on Computer Vision*, Barcelona, Spain. 18

Levoy, M. & Hanrahan, P. (1996). Light field rendering. In *Computer graphics and interactive techniques*, 31–42. 20, 66

Li, S. (2006a). Full-view spherical image camera. In *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 4, 386–390. 46, 49

Li, S. (2006b). Real-time spherical stereo. In *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 3, 1046 –1049. 48

Lothe, P., Bourgeois, S., Dekeyser, F., Royer, E. & Dhome, M. (2010). Monocular slam reconstructions and 3d city models : Towards a deep consistency. In *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, vol. 68, 201–214. 17

Lovegrove, S. & Davison, A. (2010). Real-time spherical mosaicing using whole image alignment. In *European Conference on Computer Vision*, vol. 6313, 73–86. 46, 86

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, **60**, 91–110. 14, 70

Lowe, D.G. (1991). Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 441–450. 17

Lucas, B.D. & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *International joint conference on Artificial intelligence*, 674–679. 15

Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. In *IEEE International Conference on Robotics and Automation*, 1843–1848. 15, 30

Marchand, E., Bouthemy, P. & Chaumette, F. (2001). A 2d-3d model-based approach to real-time visual tracking. *Image and Vision Computing*, **19**, 941–955. 17

Marr, D., Ullman, S. & Poggio, T. (2010). *Vision : A Computational Investigation Into the Human Representation and Processing of Visual Information*. MIT Press. 6

Martinec, D., Pajdla, T., Martinec, D. & Martinec, D. (2002). Structure from many perspective images with occlusions. In *European Conference on Computer Vision*, 355–369. 16

Mei, C. & Rives, P. (2007). Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, 3945 –3950. 43, 44

Mei, C., Benhimane, S., Malis, E. & Rives, P. (2006). Constrained multiple planar template tracking for central catadioptric cameras. In *British Machine Vision Conference*. 15

Mei, C., Sibley, G., Cummins, M., Newman, P. & Reid, I. (2010). Rslam : A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, **94**, 198–214, special issue of BMVC. 16, 86

Meilland, M., Comport, A.I. & Rives, P. (2010a). A spherical robot-centered representation for urban navigation. In *IEEE International Conference on Intelligent Robots and Systems*, 5196 –5201. 6

Meilland, M., Comport, A.I. & Rives, P. (2010b). A spherical robot-centered representation for urban navigation. In *Journee des Jeunes Chercheurs en Robotique - Journees Nationales du GDR Robotique.*. 6

Meilland, M., Comport, A.I. & Rives, P. (2011a). Dense visual mapping of large scale environments for real-time localisation. In *IEEE International Conference on Intelligent Robots and Systems*, 4242 –4248. 7

Meilland, M., Comport, A.I. & Rives, P. (2011b). Real-time dense visual tracking under large lighting variations. In *British Machine Vision Conference*, 45.1–45.11. 7

Menegatti, E., Maeda, T. & Ishiguro, H. (2004). Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, **47**, 251 – 267. 19

Mezouar, Y. & Chaumette, F. (2003). Optimal camera trajectory with image-based control. *International Journal of Robotics Research*, **22**, 781–804. 19

Mikolajczyk, K. & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 1615–1630. 70

Montemerlo, M., Thrun, S., Koller, D. & Wegbreit, B. (2002). FastSLAM : A factored solution to the simultaneous localization and mapping problem. In *AAAI National Conference on Artificial Intelligence*. 16

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F. & Sayd, P. (2006). Real time localization and 3d reconstruction. In *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 363–370. 16

Nayar, S. (1997). Catadioptric omnidirectional camera. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 482–. 43

Newcombe, R.A., andOtmar Hilliges, S.I., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S. & Fitzgibbon, A. (2011a). Kinectfusion : Real-time dense surface mapping and tracking. In *International symposium on mixed and augmented reality*. 48

Newcombe, R.A., Lovegrove, S. & Davison, A.J. (2011b). Dtam : Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision*. 16, 18, 86

Nistér, D., Naroditsky, O. & Bergen, J. (2004). Visual odometry. In *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 652–659. 16

Ogale, A.S. & Aloimonos, Y. (2005). Shape and the stereo correspondence problem. *International Journal on Computer Vision*, **65**, 147–162. 53, 55

Olson, E., Leonard, J. & Teller, S. (2006). Fast iterative alignment of pose graphs with poor initial estimates. In *IEEE International Conference on Robotics and Automation*, 2262 –2269. 65

Panin, G. & Knoll, A. (2008). Mutual information-based 3d object tracking. *International Journal of Computer Vision*, **78**, 107–118. 100

Pfeil, J., Hildebrand, K., Gremzow, C., Bickel, B. & Alexa, M. (2011). Throwable panoramic ball camera. In *SIGGRAPH Demonstration*. 47

Remazeilles, A., Chaumette, F. & Gros, P. (2004). Robot motion control from a visual memory. In *IEEE International Conference on Robotics and Automation*, vol. 4, 4695–4700. 19

Richa, R., Sznitman, R., Taylor, R. & Hager, G. (2011). Visual tracking using the sum of conditional variance. In *IEEE International Conference on Intelligent Robots and Systems*, 2953 –2958. 100, 104

Royer, E., Lhuillier, M., Dhome, M. & Chateau, T. (2005). Localization in urban environments : Monocular vision compared to a differential gps sensor. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 114–121. 17, 19, 39

Saff, E. & Kuijlaars, A. (1997). Distributing many points on a sphere. *The Mathematical Intelligencer*, **19**, 5–11. 57

Scharstein, D. & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal on Computer Vision*, **47**, 7–42. 53

Segvic, S., Remazeilles, A., Diosi, A. & Chaumette, F. (2007). Large scale vision based navigation without an accurate global reconstruction. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–8. 19

Seitz, S.M., Curless, B., Diebel, J., Scharstein, D. & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. 16, 100

Shi, J. & Tomasi, C. (1994). Good features to track. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 593–600. 15, 69

Silveira, G. & Malis, E. (2007). Real-time visual tracking under arbitrary illumination changes. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 1 –6. 99, 100, 101

Silveira, G. & Malis, E. (2010). Unified direct visual tracking of rigid and deformable surfaces under generic illumination changes in grayscale and color images. *International Journal of Computer Vision*, **89**, 84–105. 99

Silveira, G., Malis, E. & Rives, P. (2008). An efficient direct approach to visual SLAM. *IEEE Transactions on Robotics*, **24**, 969–979. 15

Slama, C.C. (1980). *Manual of Photogrammetry*. American Society of Photogrammetry, 4th edn. 24

Spinello, L. & Arras, K.O. (2011). People detection in rgb-d data. In *International Conference on Intelligent Robots and Systems*. 48

Svoboda, T., Martinec, D. & Pajdla, T. (2005). A convenient multicamera self-calibration for virtual environments. *Presence : Teleoperators and Virtual Environments*, **14**, 407–422. 49

Szalay, A.S. & Brunner, R.J. (1999). Astronomical archives of the future : a virtual observatory. *Future Generation Computer Systems*, **16**, 63–72. 57

Szeliski, R. (2006). Image alignment and stitching : a tutorial. *Foundations and Trends in Computer Graphics and Vision*, **2**, 1–104. 46, 56

Tardif, J.P., George, M., Laverne, M., Kelly, A. & Stentz, A. (2010). A new approach to vision-aided inertial navigation. In *IEEE International Conference on Intelligent Robots and Systems*, 4161 –4168. 16

Tegmark, M. (1996). An icosahedron-based method for pixelizing the celestial sphere. *The Astrophysical Journal*, **470**, L81. 57

Thrun, S. (2002). Probabilistic robotics. *ACM Communications*, **45**, 52–57. 16

Tola, E., Lepetit, V. & Fua, P. (2010). Daisy : an efficient dense descriptor applied to wide baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**, 815–830. 55, 70

Triggs, B., Mclauchlan, P.F., Hartley, R.I. & Fitzgibbon, A.W. (2000). *Bundle Adjustment – A Modern Synthesis*, vol. 1883. 16

Tsai, R.Y. (1992). Radiometry. chap. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, 221–244. 23

Tykkala, T. & Comport, A.I. (2011). A dense structure model for image based stereo slam. In *IEEE International Conference on Robotics and Automation*, 1758 –1763. 16

Vacchetti, L., Lepetit, V. & Fua, P. (2004). Stable Real-Time 3D Tracking using Online and Offline Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**, 1385–1391. 17

Vincent, L. (2007). Taking online maps down to street level. *Computer*, **40**, 118–120. 39

Viola, P. & Wells, I., W.M. (1995). Alignment by maximization of mutual information. In *International Conference on Computer Vision*, 16–23. 18, 100

Wehner, R., Michel, B. & Antonsen, P. (1996). Visual navigation in insects : Coupling of egocentric and geocentric information. *Journal of Experimental Biology*, **199**, 129–140. 5

Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I. & Tardós, J. (2009). A comparison of loop closing techniques in monocular slam. *Robotics and Autonomous Systems*. 16

Zaharescu, A., Horaud, R.P., Ronfard, R. & Lefort, L. (2006). Multiple camera calibration using robust perspective factorization. In *International Symposium on 3D Data Processing, Visualization and Transmission*, 504–511. 49

Zhang, L., Curless, B., Hertzmann, A. & Seitz, S.M. (2003). Shape and motion under varying illumination : Unifying structure from motion, photometric stereo, and multi-view stereo. In *IEEE International Conference on Computer Vision*, 618–625. 100

Zhang, W. & Kosecka, J. (2006). Image based localization in urban environments. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission*, 33–40. 19

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal on Computer Vision*, **13**, 119–152. 14

Zhang, Z. (1995). Parameter Estimation Techniques : A Tutorial with Application to Conic Fitting. Tech. Rep. RR-2676, INRIA. 31, 32

Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In *International Conference on Computer Vision*, 666–673. 23, 24

Cartographie RGB-D dense pour la localisation visuelle temps-réel et la navigation autonome

Résumé : Dans le contexte de la navigation autonome en environnement urbain, une localisation précise du véhicule est importante pour une navigation sure et fiable. La faible précision des capteurs bas coût existants tels que le système GPS, nécessite l'utilisation d'autres capteurs eux aussi à faible coût. Les caméras mesurent une information photométrique riche et précise sur l'environnement, mais nécessitent l'utilisation d'algorithmes de traitement avancés pour obtenir une information sur la géométrie et sur la position de la caméra dans l'environnement. Cette problématique est connue sous le terme de Cartographie et Localisation Simultanées (*SLAM* visuel). En général, les techniques de *SLAM* sont incrémentales et dérivent sur de longues trajectoires. Pour simplifier l'étape de localisation, il est proposé de découpler la partie cartographie et la partie localisation en deux phases : la carte est construite hors-ligne lors d'une phase d'apprentissage, et la localisation est effectuée efficacement en ligne à partir de la carte *3D* de l'environnement. Contrairement aux approches classiques, qui utilisent un modèle *3D* global approximatif, une nouvelle représentation égo-centrée dense est proposée. Cette représentation est composée d'un graphe d'images sphériques augmentées par l'information dense de profondeur (*RGB+D*), et permet de cartographier de larges environnements. Lors de la localisation en ligne, ce type de modèle apporte toute l'information nécessaire pour une localisation précise dans le voisinage du graphe, et permet de recaler en temps-réel l'image perçue par une caméra embarquée sur un véhicule, avec les images du graphe, en utilisant une technique d'alignement d'images directe. La méthode de localisation proposée, est précise, robuste aux aberrations et prend en compte les changements d'illumination entre le modèle de la base de données et les images perçues par la caméra. Finalement, la précision et la robustesse de la localisation permettent à un véhicule autonome, équipé d'une caméra, de naviguer de façon sure en environnement urbain.
Mots clés : Localisation, cartographie, suivi visuel, dense, direct, synthèse de nouvelle vue, robuste, temp-réel, *SLAM*, navigation.

Dense RGB-D mapping for real-time localisation and autonomous navigation

Abstract: In an autonomous navigation context, a precise localisation of the vehicule is important to ensure a reliable navigation. Low cost sensors such as *GPS* systems are inacurrate and inefficicent in urban areas, and therefore the employ of such sensors alone is not well suited for autonomous navigation. On the other hand, camera sensors provide a dense photometric measure that can be processed to obtain both localisation and mapping information. In the robotics community, this problem is well known as Simultaneous Localisation and Mapping (*SLAM*) and it has been studied for the last thirty years. In general, SLAM algorithms are incremental and prone to drift, thus such methods may not be efficient in large scale environments for real-time localisation. Clearly, an *a-priori* 3D model simplifies the localisation and navigation tasks since it allows to decouple the structure and motion estimation problems. Indeed, the map can be previously computed during a learning phase, whilst the localisation can be handled in real-time using a single camera and the pre-computed model. Classic global 3D model representations are usually inacurrate and photometrically inconsistent. Alternatively, it is proposed to use an ego-centric model that represents, as close as possible, real sensor measurements. This representation is composed of a graph of locally accurate spherical panoramas augmented with dense depth information. These augmented panoramas allow to generate varying viewpoints through novel view synthesis. To localise a camera navigating locally inside the graph, we use the panoramas together with a direct registration technique. The proposed localisation method is accurate, robust to outliers and can handle large illumination changes. Finally, autonomous navigation in urban environments is performed using the learnt model, with only a single camera to compute localisation.
Keywords: Localisation, mapping, visual tracking, dense, direct, novel view synthesis, robust, real-time, *SLAM*, navigation.

MINES ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY